

Joona Kivelä

HOITOTASOJEN SUUNNITTELUAUTOMAATTI

Opinnäytetyö

CENTRIA-AMMATTIKORKEAKOULU

Kone- ja Tuotantotekniikan koulutusohjelma

Kesäkuu 2016

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Kokkola	Aika Kesäkuu 2016	Tekijä/tekijät Joona Kivelä
Koulutusohjelma Kone- ja Tuotantotekniikka		
Työn nimi HOITOTASOJEN SUUNNITTELUAUTOMAATTI		
Työn ohjaaja Mika Kumara	Sivumäärä 38	
Työelämäohjaaja Tuomas Jokioinen		
<p>Tämän opinnäytetyö tarkoituksena oli suunnitella ja valmistaa hoitotasojen suunnitteluautomaatti Merius Oy:lle. Merius Oy tuottaa suunnittelu- ja projektinhallinta palveluita teollisuudelle. Suunnitteluautomaatin tarkoitus oli vähentää yksinkertaista ja toistuvaa työtä. Työn tuli myös täyttää standardien ja lain asettamat vaatimukset, jotka koskevat hoitotasoja.</p> <p>Tässä opinnäytetyössä käydään läpi, miten itse tässä tapauksessa olen toteuttanut suunnitteluautomaatin, joten se ei ole suora ohje, siihen miten suunnitteluautomaatti tehdään. Suunnitteluautomaatti voidaan jakaa kolmeen eri osaan mallintamiseen, sääntöjen kirjoittamiseen ja kaavakkeen tekoon.</p> <p>Opinnäytetyön teoriaosassa käsitellään standardin ja lain asettamia vaatimuksia. Opinnäytetyössä käydään läpi myös luovan työn menetelmiä.</p> <p>Opinnäytetyön lopputuloksena syntyi toimiva suunnitteluautomaatti Merius Oy:n käyttöön. Työn lopussa pohditaan, mitä työssä olisi voitu tehdä toisin ja missä olisi parannettavaa.</p>		

Asiasanat Suunnitteluautomaatti, Hoitotaso, Inventor, iLogic

ABSTRACT

CENTRIA UNIVERSITY OF APPLIED SCIENCES Kokkola	Date June 2016	Author Joona Kivelä
Degree programme Mechanical and Production Engineering		
Name of thesis INDUSTRIAL CATWALK DESIGN AUTOMAT		
Instructor Mika Kumara		Pages 38
Supervisor Tuomas Jokioinen		
<p>The purpose of this thesis was to design and produce industrial catwalk design automat for Merius Oy. Merius Oy produces engineering and project management services for the industry. Industrial catwalk design automat purpose was to reduce the simple and the repetitive work. Industrial catwalk design automat had to also meet the requirements of standards and laws.</p> <p>In this thesis I explain how I executed this design automat, because of that it's not instruction how to make design automat. Design automat can be divided into three parts modeling, coding and form making.</p> <p>The theoretical part of this thesis contains information of standards and laws that controls industrial catwalks. This thesis also goes through some creative work methods.</p> <p>The final result of the thesis was a functional design automat for Merius Oy use. At the end of thesis, I cover what could be done differently and what could be improved.</p>		

Key words

Design automat, Industrial catwalk, Inventor, iLogic

KÄSITTEIDEN MÄÄRITTELY

Blokki	Valmis geometria, joka voidaan tuoda sketchiin
Component pattern	Toiminto, jolla voidaan kopioida osaa tietyllä välillä
Content center	Inventorin osakirjasto
Level of detail	Määrittää, voidaanko osa deaktivoida kokoonpanossa
Multi body-mallinnus	Osatiedoston sisään on mallinnettu useita toisistaan riippumattomia osia
Sculpt	Työkalu, jolla voidaan leikata tai lisätä materiaalia kappaleeseen
Sketch	Inventorin piirrettä ohjaava geometria
Solid body	Osan runko
Surface	Piirustuksesta pursotettava pinta

TIIVISTELMÄ

ABSTRACT

KÄSITTEIDEN MÄÄRITTELY

SISÄLLYS

1 JOHDANTO	1
2 MERIUS OY.....	2
3 LUOVA TYÖ	3
4 INVENTOR.....	6
5 HOITOTASOT JA SUOJAKAITEET	9
5.1 Hoitotason vaatimukset	9
5.2 Suojakaiteiden vaatimukset	11
6 SUUNNITTELUAUTOMAATTI.....	15
6.1 Mallintaminen	15
6.1.1 Runko	16
6.1.2 Kaiteet	17
6.1.3 Runkopalkit	18
6.1.4 Tukipalkit.....	20
6.2 Kokoonpanot	21
6.3 Säännöt.....	22
6.3.1 Parametrien linkitys.....	23
6.3.2 Runko	24
6.3.3 Tukipalkit.....	26
6.3.4 Kaiteet	27
6.3.5 Potkulista	31
6.3.6 Muita sääntöjä.....	32
6.4 Kaavake.....	34
7 POHDINTA	35
LÄHTEET	38

KUVAT

KUVA 1. Konsultointi-, suunnittelu- ja projektinhallintapalvelut	2
------------------------------------------------------------------------	---

KUVA 2. Luovan työn vaiheet	5
KUVA 3. Parameters	6
KUVA 4. iLogic käyttöliittymä	7
KUVA 5. Mittojen keskitys	8
KUVA 6. Kulkuteiden käyttöalueet.....	9
KUVA 7. Suojakaide	12
KUVA 8. Kahden suojakaiteen väliin jäävä vapaa tila.....	13
KUVA 9. Portaissa oleva suojakaide	14
KUVA 10. Skeleton-malli.....	16
KUVA 11. Runko ja ritilä	17
KUVA 12. Käsijohteen sketch ja pursotus	18
KUVA 13. Runkopalkin pursotus	19
KUVA 14. Rungon aputasot	19
KUVA 15. Surfacen luonti.....	20
KUVA 16. U-tukipalkin sculpt	21
KUVA 17. Kokoonpano	22
KUVA 18. Tärkeitä parametreja ja niiden käyttö kohteet	23
KUVA 19. Parametrien linkitys.....	24
KUVA 20. Palkkityyppi sääntö	25
KUVA 21. U-Palkin suuntaa muuttava sääntö	26
KUVA 22. Tukipalkin deaktivoiva sääntö.....	27
KUVA 23. Pystyputkien määrän sääntö	27
KUVA 24. Kaiteen etäisyyttä reunasta muuttava sääntö	28
KUVA 25. Kaiteen puolen sääntö.....	29
KUVA 26. Kaide elementin deaktivointi.....	29
KUVA 27. Kaide elementin deaktivointi.....	29
KUVA 28. Välitolppien ja välijohteiden deaktivointi	30
KUVA 29. Välitolpan pursotusta muuttava sääntö.....	30
KUVA 30. Kaiteen profiilin sääntö	31
KUVA 31. Potkulistan sääntöjä	32
KUVA 32. Event trigger	33
KUVA 33. Kaavake	33
KUVA 34. Kaavake editori.....	34

1 JOHDANTO

Toimeksiannon tähän työhön sain Merius Oy:ltä, jonka toimipiste on Kokkolassa. Merius Oy:n suurin liikevaihto tulee investoivasta suurteollisuudesta. Suunnittelussa syntyy paljon asioita, jotka voitaisiin automatisoida. Hoitotasot ovat yleensä suoria tasoja, joita käytetään kulkuteinä laitteille. Tästä syystä ne ovat lähtökohtana helposti automatisoitavissa. Työn aloituspalaverissa tavoitteeksi asetettiin, että tuotaisin suunnitteluautomaatit sekä portaille että hoitotasaille. Tämä tavoite oli erittäin optimistinen, ja sitä jouduttiinkin tiivistämään huomattavasti työn edetessä. Työn laajuus rajattiin lopulta kattamaan ainoastaan hoitotaso.

Tässä opinnäytetyössä oli tavoitteena suunnitella ja valmistaa hoitotasojen suunnitteluautomaatti. Ennen työn aloitusta materiaalinhakuvaiheessa totesin, että työhön liittyvää teoreettista tietoa oli saatavilla erittäin vähän. Yksi syy tähän on, että suunnitteluautomaatteja valmistavat yritykset eivät halua lisää kilpailua, eivätkä siten julkaise oppaita. Merius Oy kuitenkin järjesti koulutuksen, jossa työn suorittamiselle luotiin hyvä perusta. Työn suorittaminen vaati silti itseopiskelua iLogic:n ja Visual Basicin osalta.

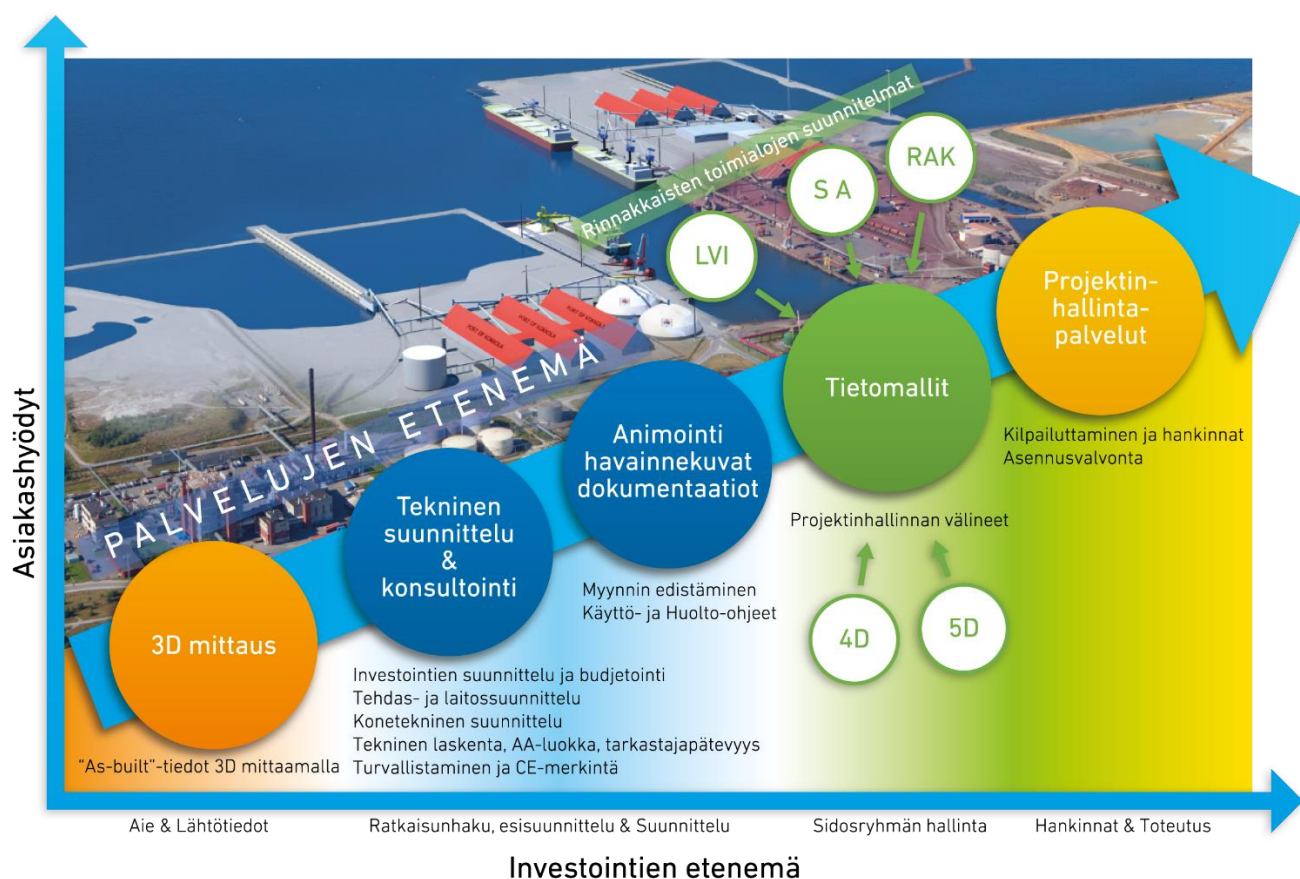
Opinnäytetyössä perehdytään aluksi luovan työn tekniikkoihin, jossa käydään läpi, millä tavalla luovaa työtä voidaan tehostaa. Sen jälkeen tutustutaan Inventoriin ja käydään läpi iLogic:n toimintaa. Tämän jälkeen käydään läpi mitä vaatimuksia kaiteille ja hoitotasaille on standardissa ja laissa määritetty. Itse suunnitteluautomaatin teoria on jaettu neljään osaan. Ensimmäisessä käydään läpi, miten hoitotaso ja kaiteet mallinnetaan. Toisessa selitetään mitä kokoonpanot sisältävät. Kolmannessa käydään läpi suunnitteluautomaattia ohjaavia sääntöjä ja kerrotaan, miten ne on tehty. Neljännessä käydään lyhyesti kaavakkeeseen liittyviä asioita.

Työn suorittaminen oli vaativa prosessi, mutta samalla myös todella mielenkiintoinen. Merius Oy antoi minulle mahdollisuuden suorittaa työ heidän toimipisteessään, ja tämä helpotti huomattavasti työn etenemistä, koska pystyin keskustelemaan ongelmista ja työstä ohjaajan kanssa heti.

2 MERIUS OY

Merius Oy on Kokkolaan perustettu suunnittelu-, konsultointi- ja projektihallintapalveluita tarjoava yritys. Se on perustettu vuonna 2003. Merius tuottaa kuvassa 1 näkyviä konsultointi-, suunnittelu- ja projektihallintapalveluita. Vuonna 2016 Merius Oy:n palveluksessa työskenteli 16 henkilöä. Merius Oy hyödyntää ennakkoluulottomasti uusia toimintatapoja ja teknologiaa tekemisessään. Merius Oy on panostanut huomattavasti 3D-mittauspalveluihin ja niiden hyödyntämiseen toiminnassa. Se on tällä hetkellä yksi suomen parhaiten 3D-mittausta hyödyntävä yritys.

Merius Oy:n toiminnan lähtökohtana on asiakkaan tarpeiden täyttäminen. Merius Oy hyödyntää Lean-ajattelua toiminnassaan ja käyttää tuottamiaan palveluita asiakkaan tarpeiden mukaan, samalla tuottaen korkealaatuisen ja toimitusvarman kokonaisuuden. Kuva 1 havainnollistaa tätä tapahtumien sarjaa.



KUVA 1. Konsultointi-, suunnittelu- ja projektihallintapalvelut (Merius Oy)

3 LUOVA TYÖ

Luovuus on taito, jonka kuka tahansa voi oppia harjoittelun kautta. Luovuus on kyky rikkoa ajatteluprosesseja ja synnyttää uusia. Se ei ole kokoelma tekniikoita ja ohjeita, vaan tietynlaista valmiutta toimintaan. Toisilla on suurempi taipumus luovaan ajatteluun ja toimintaan. Vain harva on luova perimmiltään. (FilosofianAkatemia 2009, 4.)

Luova työ koostuu muutamasta erilaisesta työvaihteesta. Luonteeltaan nämä työvaiheet ovat hajottavia tai kokoavia. Hajottavalla eli divergentillä toiminnalla tarkoitetaan uuden etsimistä ja siinä pyritään synnyttämään uusia tapoja ajatella, olla ja toimia. Kokoavalla eli konvergentillä toiminnalla pyritään hioamaan tiettyä yksilöityä toimintaa. Vanhaa ja hyväksi havaittua toimintaa pidetään myös konvergenttina toimintana. (FilosofianAkatemia 2009, 5.)

Luova työ voidaan jakaa viiteen eri työvaiheeseen:

1. Kerääminen
2. Valikointi
3. Luonnostelu
4. Jalostaminen
5. Viimeistely

Keräämisvaiheessa etsitään ja synnytetään uusia ideoita mahdollisimman paljon. Yksi tapa on ideoiden poimiminen. Tämä tapahtuu kirjaamalla kaikki ideat talteen, keräämällä mielenkiintoiset artikkelit ja internetsivut talteen. Toinen tapa on tekemällä tutkimusta. Tällä tavalla kerätään järjestelmällisesti materiaalia liittyen työhön. Tutkimuksen ei tarvitse tuottaa suoraa tutkimustulosta, vaan itse tutkimuksen tulos on luova teos. Kolmas tapa on improvisoida. Improvisaatio vaatii harjaantuneisuutta. Harjaantuneisuuttasi osaat luoda uusia ajatuksia ja ideoita työhösi. Itse kriittisyys kannattaa unohtaa tai jättää vähäiseksi keräämisvaiheessa. Keräämisvaiheessa on erittäin tärkeää lajitella kerätty materiaalin. (FilosofianAkatemia 2009, 6–12.)

Luovan työn valikointivaiheessa ruvetaan jalostamaan keräämisvaiheessa tuotettua materiaalia. Ensimmäisenä on tärkeää käydä materiaali läpi ja valita käyttökelpoinen materiaali työhön. Valikointi vaiheessa kritiikki materiaalia kohtaan on suotavaa. Yksi tapa seuloa materiaalia on dialektinen ja dialogi-

nen metodi. Tällä metodilla arvioidaan materiaalin paremmuutta toisiinsa nähden ja pyritään sovittamaan materiaalista sopusointuinen kokonaisuus. Tällä metodilla materiaalin joukosta voi löytyä usein joukko erittäin hyviä ideoita tai jopa paras idea. Toinen tapa on luokitella materiaali erilaisiin luokkiin. Luokittelussa on jopa suotavaa luoda ihan uusia kategorioita esim. pyörivät asiat. (FilosofianAkademia 2009, 13–15.)

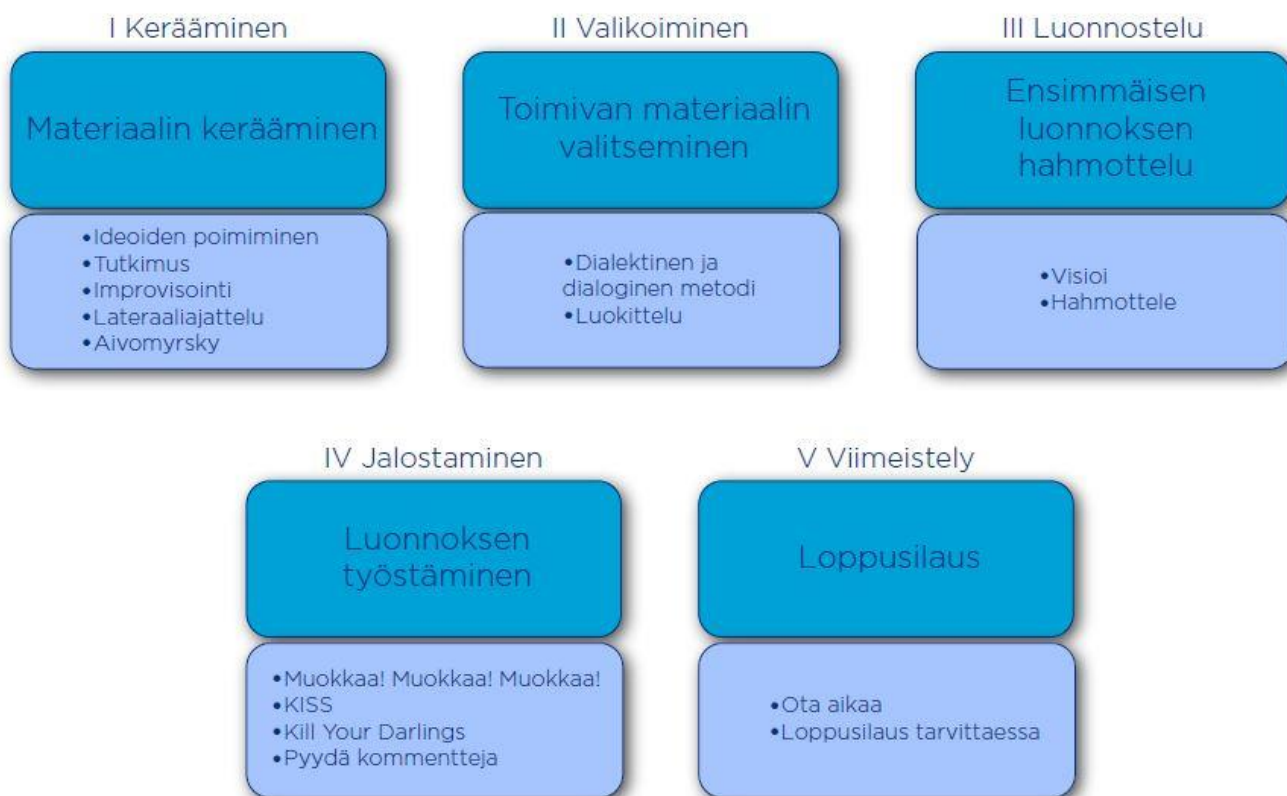
Luonnosteluvaiheessa ollaan ensimmäistä kertaa itse työn kanssa kosketuksissa. Tästä syystä se on yksi luovan työn suurimpia vaiheita. Suurimpia virheitä luonnosteluvaiheessa on, että kuvitellaan olevan pian jalostamis- tai viimeistelyvaiheessa. Yksi luonnostelun tapa on visiointi. Siinä hyödynnetään mielikuvi- tusta ja pyritään luomaan mahdollisimman valmis kuva lopullisesta työstä. Mitä paremmin lopullinen työ voidaan kuvitella, sen helpompaa sen tekeminen on. Toinen tapa on hahmottelu. Tällä menetelmällä työstä luodaan raakaversio. Sen tarkoitus on olla suuntaa antava. Hahmotelmassa ei pidä keskittyä liikaa yksityiskohtiin, vaan luoda suuria linjoja kohti lopullista työtä. (FilosofianAkademia 2009, 15–17.)

Jalostamisvaiheessa työtä tarkastellaan kriittisesti kokonaisuutena, josta poistetaan osat jotka eivät sovi siihen. Jalostaminen voidaan aloittaa monella tapaa; näistä yksin on muokkaaminen. Muokkaamisella tarkastellaan työtä kriittisesti ja samalla hiotaan sen yksityiskohta kokonaisuuteen sopiviksi. Siinä myös korjataan virheet, jotka eivät ole tarkoituksenmukaisia. Muokkaamisen tarkoitus on saattaa työ aina valmiimmaksi kokonaisuudeksi. Toinen tapa on KISS (Keep it Simple, Stupid), eli jätetään kaikki turha pois työstä. Siinä työstä jätetään pois kaikki, mikä voidaan jättää pois. Ideana on ilmaista asia mahdollisimman yksinkertaisesti.

Kaikki tulee tehdä niin yksinkertaisesti kuin mahdollista – muttei yksinkertaisemmin.
Albert Einstein

Kolmas tapa on erittäin lyhyt ja ytimekäs. Sitä kutsutaan ”Kill Your Darlings” -tavaksi. Sen periaate on tarkastella työtä kokonaisuutena, ja jos jokin yksityiskohta vie huomion kokonaisuudelta, on tämä yksityiskohta poistettava työstä. Tämä ei kuitenkaan tarkoita, että yksityiskohta olisi huono. Tällä saadaan luotua työtä kokonaisuus, joka ei koostu yksittäisistä neronleimuksista. (FilosofianAkademia 2009, 18–22.)

Viimeistelyvaiheessa työ saatetaan päätöksen ja työn jalostaminen päätetään, jotta tähän päätökseen päästään on työtä hyvä tarkastella ulkopuolisen silmin. Tämä auttaa näkemään kokonaisuuden uudella tavalla. Loppusilaus tehdään, jos työstä löytyy jotain, mitä halutaan vielä muuttaa. Loppusilauksessa on tärkeää, ettei siitä ala loputtoman hiomisen kierre. (FilosofianAkademia 2009, 22–24.)



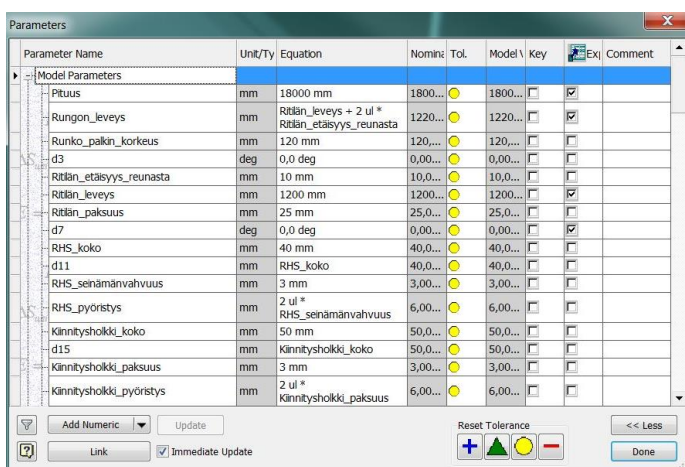
KUVA 2. Luovan työn vaiheet (Filosofian Akatemia 2009)

4 INVENTOR

Inventor on yksi Autodeskin tuottama suunnitteluohjelma, joka on osa suurta tuoteperhettä (Autodesk.com). Inventor on 3D-suunnitteluohjelma, jolla pystytään valmistamaan tarkoista mittakuvista aina 3D-animaatioihin saakka. Sillä pystytään myös suorittamaan lujuulaskennan analyyssejä. Autodesk julkaisi Inventorin ensimmäisen kerran vuonna 1999 syyskuussa nimellä Inventor 1. Tämän jälkeen Inventor:sta on julkaistu uusi versio joka vuosi. Uusin Inventorin versio kantaa nimeä Inventor 2017. (Autodesk Inventor Release History 2015.)

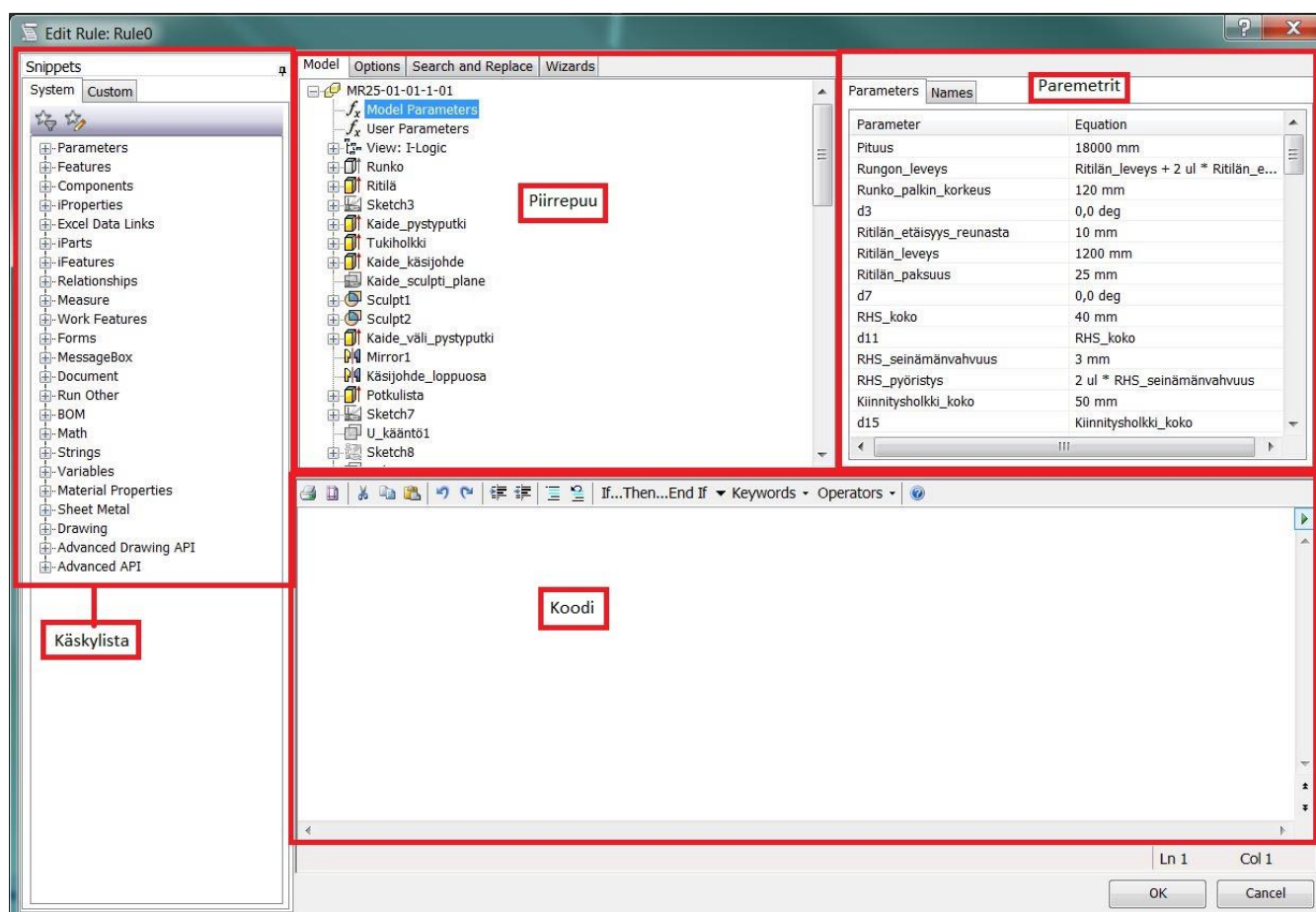
ILogic on julkaistu Inventorin 2011 versiossa. ILogic on Inventor-ominaisuus, jolla voidaan automatisoida ja hallita Inventorin ominaisuuksia. Ominaisuuksien muuttaminen ja hallitseminen perustuvat pääasiassa parametrien muuttamiseen.

Parametrilla voidaan käsitellä 2D-sketchiin mitoitettua mittaa tai piireteeseen annettua mittaa/lukuarvoa. Kaikki osaan syötetyt mitat löytyvät parametris-ikkunasta. Kuvassa 3 näkyy, miltä parametrit ikkuna näyttää. Inventor nimeää parametrit automaattisesti d1, d2, d3 ja niin edelleen. Tästä syystä käyttäjän kannattaa itse nimetä jokainen parametri. Tämä voidaan ja kannattaakin tehdä samalla, kun mittaa tai piirrettä asetetaan paikalleen. Parametrin nimeäminen tapahtuu kirjoittamalla nimi = parametri, esim. Pituus = 500mm. Parameters-ikkunassa käyttäjä voi itse luoda myös valmiita parametreja, joita voi käyttää myöhemmin mitoissa tai piirteissä. Parametreista voi tehdä multi-valueja, eli tällöin parametri voi sisältää useita parametreja.



KUVA 3. Parameters (Autodesk Inventor 2014)

ILogic-käyttöliittymä on erittäin yksinkertainen, se sisältää 4 kenttää. Kuvassa 4 on esitetty nämä kentät. Käskylistasta löytyy valmiita käskyjä, joilla voidaan ohjata esim. sitä, minkä värinen jokin osa on. Piirrepuusta löytyy aktiivisena olevan osan tai kokoonpanon piirteet ja parametrit. Kun piirrepuusta valitaan jokin kohta, näkyy kyseisen kohdan parametrit parameters-kentässä. Koodikenttään kirjoitetaan koodia, jolla ohjataan osaa tai kokoonpanoa. Yksinkertaisimmillaan koodi on osan aktivoimista ja deaktivoimista, kun jokin parametri ylittää sallitun arvon. Piirrepuusta voidaan poimia piirteitä ja parametreja koodiin helposti valitsemalla se hiiren oikealla painikkeella ja valitsemalla Capture Current State. Tämä voidaan myös tehdä parameters-kentästä, jolloin voidaan valita yksittäinen parametri piirteen sisältä.



KUVA 4. iLogic käyttöliittymä (Autodesk Inventor 2014)

iLogic ymmärtää hyvin pitkälle yksinkertaista Visual Basic -koodia, joten se on aloittelevalla iLogic:n käyttäjälle suhteellisen helppoa ymmärtää ja kirjoittaa. Visual Basicin ansiosta iLogic:lla voidaan lukea ja kirjoittaa HTML, teksti, word, excel ja monia muita tiedostotyyppisiä. Se myös mahdollistaa kommunikoinnin Windowsin muiden ohjelmien kanssa. Visual Basicilla voidaan esim. tallentaa mittakuvat pdf-muodossa automaattisesti tiettyyn kansioon, kun tiedosto suljetaan. Visual Basicin parhaimpia puolia on

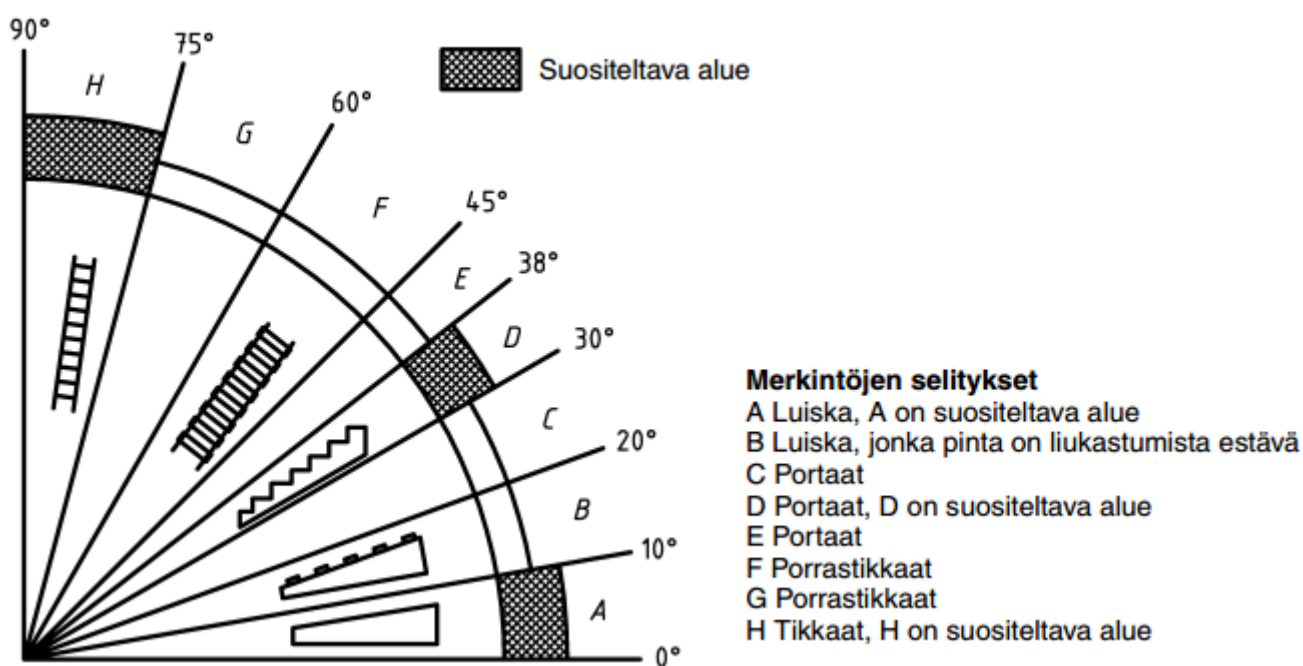
sen helppokäyttöisyys ja ohjeiden määrä. Kuvassa 5 on mitat keskittävä sääntö. iLogic:lla voidaan ajaa DLL tiedostoja, jotka on kirjoitettu täysin Visual Basicilla. Tämä vaatii laajempaa tietämystä Visual Basicista

```
Public Sub CenterAllDimensions()  
    ' Set a reference to the active drawing document  
    Dim oDoc As DrawingDocument  
    Set oDoc = ThisApplication.ActiveDocument  
  
    ' Set a reference to the active sheet  
    Dim oSheet As Sheet  
    Set oSheet = oDoc.ActiveSheet  
  
    Dim oDrawingDim As DrawingDimension  
  
    ' Iterate over all dimensions in the drawing and  
    ' center them if they are linear or angular.  
  
    For Each oDrawingDim In oSheet.DrawingDimensions  
        If TypeOf oDrawingDim Is LinearGeneralDimension Or _  
            TypeOf oDrawingDim Is AngularGeneralDimension Then  
            Call oDrawingDim.CenterText  
        End If  
    Next  
End Sub
```

KUVA 5. Mittojen keskitys (Autodesk Inventor 2014)

5 HOITOTASOT JA SUOJAKAITEET

Konedirektiivi 2006/42/EY määrittelee yleisellä tasolla vaatimukset hoitotasoilte, portaille ja suojakaitteille. Tämä on kuitenkin niin yleisellä tasolla, että sen perusteella ei voida valmistaa mitään kyseisistä tuotteista. (Konedirektiivi 2006/42/EY.) Tämän vuoksi on tehty Standardi SFS-EN ISO 14122, joka määrittelee hoitotasojen, portaiden ja suojakaitteiden tarkempia vaatimuksia. Kuvassa 6 on esitetty erilaisten kulkuteiden käyttöalueet. (SFS-EN ISO 14122-1.)



KUVA 6. Kulkuteiden käyttöalueet

5.1 Hoitotason vaatimukset

Hoitotaso on taso, jota käytetään kulkuun ja laitteiden huoltamiseen. Tämän vuoksi hoitotasot on suunniteltava ja valmistettava niin, että niiden käyttäminen on turvallista. Hoitotasolla on oltava pysyvät liukastumisen estävät ominaisuudet. Käyttäjän kanssa mahdollisesti kosketuksiin joutuvat kohdat on suunniteltava ja rakennettava niin, etteivät ne suojaavat vammoilta. Kulku hoitotasolta on suunniteltava niin, että sieltä on pystyttävä poistumaan tai evakuoimaan nopeasti hätätilanteen sattuessa. Suojakaiteet ja tukirakenteet on suunniteltava ja sijoitettava niin, että niitä käytettäisiin vaistomaisesti. (SFS-EN ISO 14122-2.)

Hoitotasot on sijoitettava mahdollisuuksien mukaan sellaisessa paikassa, etteivät ne ole haitallisten materiaalien tai aineiden päästökohdissa tai paikassa jossa hoitotasolle voi kertyä liukastumista aiheuttavaa ainetta, esimerkiksi multaa. Kun lähettyvillä on liikkuvia osia, suojaamattomia jännitteisiä sähkölaitteita, suojaamattomia kuumia pintoja on standardin EN 294 -mukainen turvaetäisyys täyttyvä. (SFS-EN ISO 14122-2.)

Hoitotaso on sijoitettava niin, että sillä työskentelevän on voitava tehdä työnsä ergonomisesti hyödyllisessä asennossa. Hoitotaso on sijoitettava mahdollisuuksien mukaan niin, että työkohteet ovat 500–1700 mm korkeudella tason pinnasta. (SFS-EN ISO 14122-2.)

Kunnossapitoon tai koneen käyttöön tarkoitetun hoitotason vapaa leveys ja pituus on määriteltävä seuraavien tekijöiden perusteella:

- Tasolla tehtävän työn vaatimat tarpeet, esim. voimankäyttö, vapaa tila liikkeille jne.
- Kuljetetaanko tasolla tavaroita
- Kuinka monta työskentelijää tasolla on yhtäaikaista
- Tasolla tehtävän työn kesto ja sen käytön taajuus
- Loukkaantuneen henkilön kuljettaminen tasolta
- Taso päättyy umpikujaan
- Lähellä olevat seinät, jotka mahdollisesti likaavat tai vahingoittavat käyttäjien vaatteita.

Hoitotason tärkeimpiä mittoja ovat vapaa korkeus ja tason leveys. Tämän vuoksi mittojen määrittämiseen hyödynnetään ihmisen antropometrisiä mittoja. Standardeissa SFS-EN 547-1 ja SFS-EN 547-3 esitettyjen arvojen mukaan vapaa korkeus on oltava vähintään 2100 mm ja leveydessä vähintään 600 mm. Hoitotason leveydeksi kuitenkin suositellaan käytettävän vähintään 800 mm. Jos normaalissa käytössä hoitotasoa käyttää useampi henkilö samanaikaisesti on hoitotason leveyden oltava vähintään 1000 mm. Hoitotason suositus mitoista voidaan poiketa, jos ympäristön tai koneen aiheuttamien rajoitusten takia. Tällöin hoitotason vapaata leveyttä ja korkeutta voidaan pienentää korkeintaan 500 mm ja 1900 mm, jos tasoa käytetään satunnaisesti tai tehtävä pienennys on lyhyellä matkalla. Jos vapaata leveyttä tai korkeutta rajoittaa jokin seinässä tai katon alapuolella oleva este, on se suojattava tapaturman estämiseksi esim. pehmusteella ja merkittävä varoituskilvellä. (SFS-EN ISO 14122-2.)

Hätäpoistumistienä käytettävän hoitotason on täytettävä viranomais määräykset. Tämänhetkisen määräyksen mukaan hätäpoistumistien leveyden on oltava vähintään 900 mm ja vapaan korkeuden 2100

mm, vähimmäiskorkeuden alapuolella ei saa olla mitään esteitä. (Ympäristöministeriön asetus rakennusten paloturvallisuudesta. 6.4.2011.)

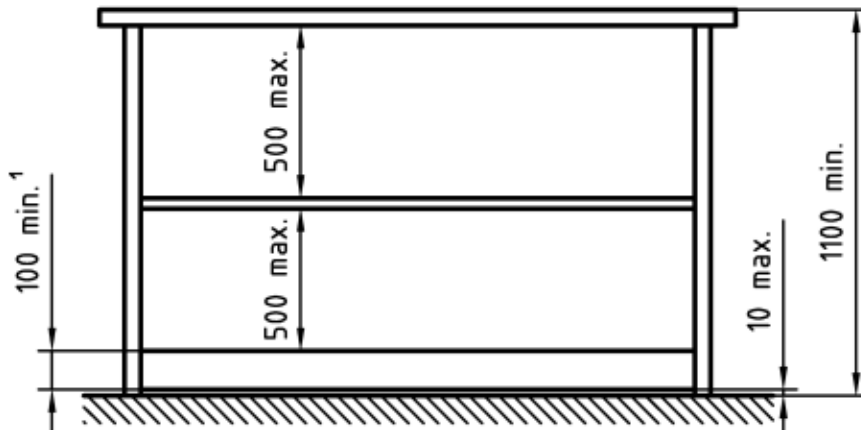
Jos hoitotasolta on mahdollista pudota 500 mm korkeudelta, on hoitotaso tällöin varustettava standardin SFS-EN ISO 14122-3 mukaisilla suojakaiteilla, jotka suojaavat putoamisen. Suojakaiteet vaaditaan myös sellaisissa paikoissa, joissa on vaarana rakenteen pettäminen tai sen uppoaminen. (SFS-EN ISO 14122-2.)

Hoitotason lattiapinnan ominaisuuksien on oltava liukastumisvaaraa vähentävä. Lattiapinnan reunan ja viereisen rakenteen välin ollessa suurempi kuin 30 mm on hoitotason reunalle asennettava jalkalista. Jos hoitotason lattiapinta on valmistettu irrotettavista osista, on osien liikkumisen oltava estetty ja kiinnitys kohdat on oltava mahdollista tarkastaa. (SFS-EN ISO 14122-2.)

Hoitotason teknisissä tiedoissa on ilmoitettava suunnittelukuorma. Hoitotason suunnittelussa on otettava huomioon vähimmäiskuormat. Tasaisesti rakenteeseen kohdistuva 2kN/m^2 :n kuormitus. 1.5kN :n kuorma, joka kohdistuu lattiarakenteen epäedullisimpaan kohtaan jakautuneena $200\text{ mm} \times 200\text{ mm}$ alueelle. Suunnittelukuormalla kuormitetun lattiapinnan taipuma ei saa ylittää $1/200$ osaa jännevälistä, eikä kuormitetun ja kuormittamattoman lattiapinnan välinen korkeusero saa ylittää 4 mm . (SFS-EN ISO 14122-2.)

5.2 Suojakaiteiden vaatimukset

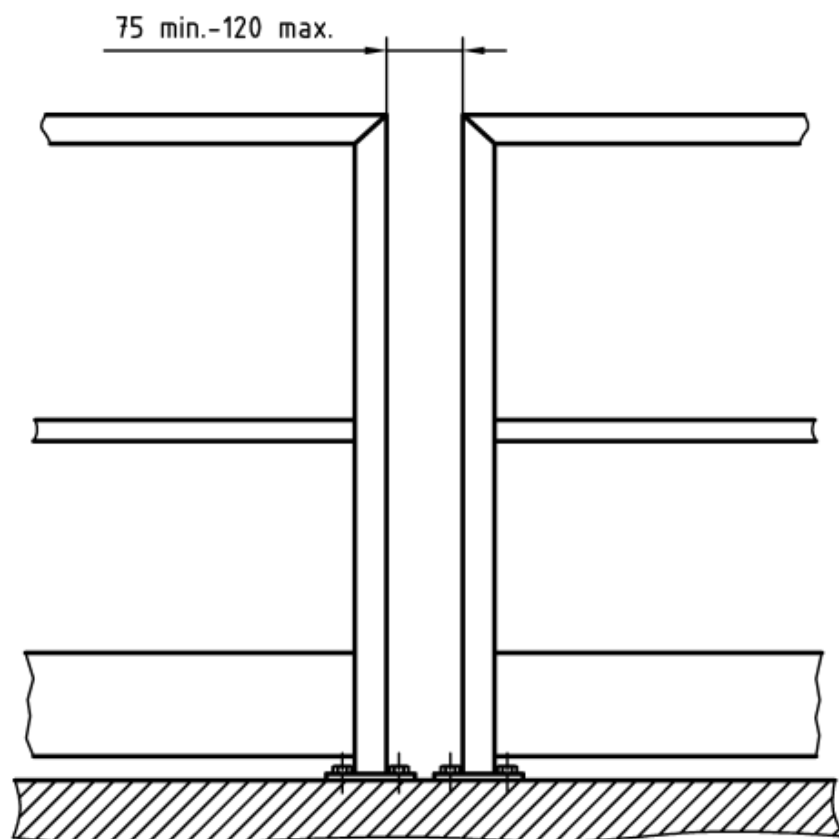
Suojakaide on oltava, jos putoamismatka on suurempi kuin 500 mm . Suojakaide on oltava vaarallisten alueiden puolella, kun hoitotason ympäröivään tilaan on mahdollista pudota tai upota. Kun hoitotason ja viereisen rakenteen välinen aukko on yli 200 mm , vaaditaan suojakaide. Kuvassa 7 on esimerkki vaakasuorasta suojakaiteesta. Jos viereinen rakenne suojaa suojakaidetta paremmin putoamiselta ei suojakaidetta vaadita. Kun hoitotason ja viereisen rakenteen välinen aukko on suurempi kuin 30 mm , vaaditaan jalkalista. (SFS-EN ISO 14122-3.)



KUVA 7. Suojakaide

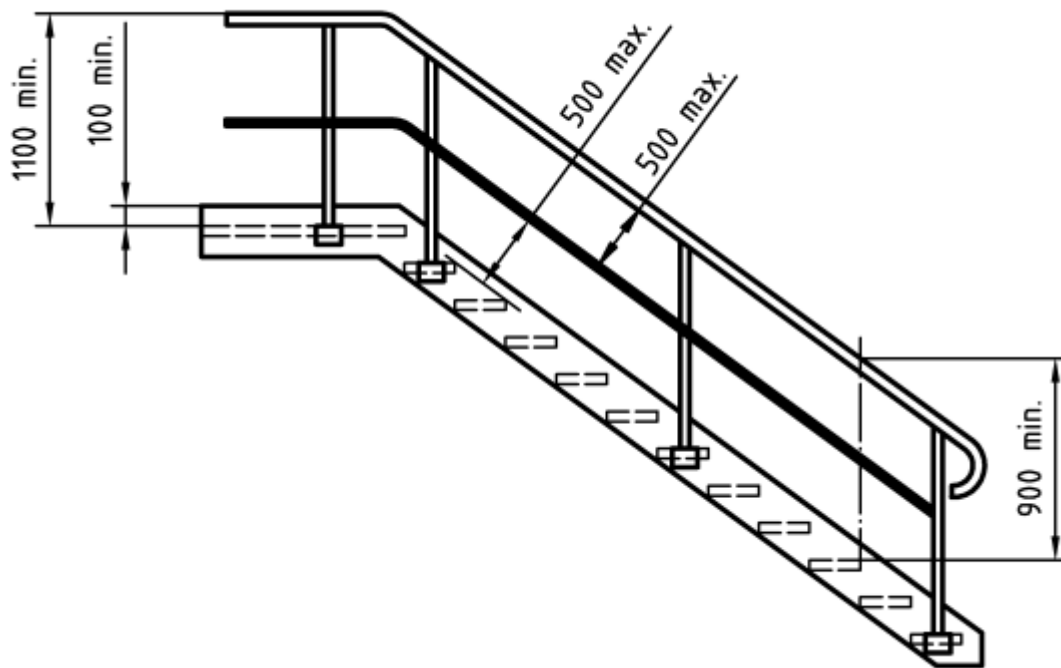
Suojakaiteen käsijohteen tulee olla vähintään 1100 mm korkea. Käsijohteiden päät eivät saa aiheuttaa vaaraa käyttäjälle, eikä käyttäjien vaatteille. Käsijohteen halkaisijaksi suositellaan 25–50 mm tai sen on oltava muuten sopivan muotoinen hyvän otteen takaamiseksi. Käsijohteen ympärille tulisi jättää vapaata tilaa 100 mm:n säteelle, lukuun ottamatta suojakaiteen rakennetta. Suojakaiteessa tulee olla vähintään jokin suojaa antava rakenne esim. välijohte. Jos suojakaiteessa käytetään välijohtetta, ei niiden välinen etäisyys saa ylittää 500 mm. Jos suojakaiteessa käytetään pystypienoja välijohteen sijasta, saa pystypienojen välinen vaakasuora etäisyys olla enintään 180 mm. Suojakaiteeseen asennetun jalkalistan etäisyys hoitotason kävelytasosta saa olla korkeintaan 10 mm. Jalkalistan korkeudeksi suositellaan vähintään 100 mm. (SFS-EN ISO 14122-3.)

Kaidetolppien välinen etäisyys on korkeintaan 1500 mm. Jos käytetään pidempää väliä, on kiinnitettävä huomiota kaidetolppien kiinnityskohtien lujuuteen ja kiinnitysvälineisiin. Kun suojakaide ei ole yhtenäinen ja käsijohteiden väliin jää aukko, on aukon oltava leveydeltään vähintään 75 mm ja enintään 120 mm, tällä estetään takertuminen. Jos aukko ylittää 120 mm leveyden, on siihen asennettava itsestään sulkeutuva portti. Kuvassa 8 on esimerkki suojakaiteiden väliin jäävästä aukosta. (SFS-EN ISO 14122-3.)



KUVA 8. Kahden suojakaiteen väliin jäävä vapaa tila

Portaissa tulee olla vähintään yksi käsijohde, kun portaiden leveys on yli 1200 mm on käsijohde oltava molemmilla puolilla. Jos portaiden nousukorkeus on suurempi kuin 500 mm ja kun portaiden sivupalkkien vieressä on yli 200 mm leveä aukko, on aukon puolella oltava suojakaide. Portaissa olevan suojakaiteen käsijohteen pystysuora etäisyys askelreunasta on oltava 900–1000 mm. Tasanteella olevaan suojakaiteeseen pätee vaakatasossa olevan suojakaiteen mitat. Kuvassa 9 on esimerkki portaiden suojakaiteesta ja siitä kuinka ne jatkuvat vaakasuoraksi suojakaiteeksi. (SFS-EN ISO 14122-3.)



KUVA 9. Portaissa oleva suojakaide

Suojakaiteen rakenteellisia ominaisuuksia koskevat seuraavat vaatimukset. Kun suojakaiteeseen kohdistetaan vaakasuunnassa oleva pistekuorma, on suojakaiteen kestävä se ilman havaittavia pysyviä muodonmuutoksia. Tämä pistekuorma on yhtä suuri tai suurempi kuin käyttökuormitus, pistekuorma kohdistetaan kaidetolpan yläpäähän ja sen jälkeen käsijohteen epäedullisimpaan kohtaan. Edellä esitetyissä tapauksissa taipuma ei saa ylittää 30 mm kuormitettuna. Vähimmäiskuormituksena käytetään vähintään $F_{\min}=300 \text{ N/m} \times \text{suurin kahden peräkkäisen kaidetolpan välinen etäisyys, mitattuna kaidetolpan keskiakselista. (SFS-EN ISO 14122-3.)}$

6 SUUNNITTELUAUTOMAATTI

Suunnitteluautomaatin tarkoitus on vähentää yksinkertaisten tasojen mallintamista ja piirustusten luomista. Suunnitteluautomaatin valmistus kannattaa aloittaa suunnittelemalla suunnitteluautomaatti niin pitkälle kuin etukäteen voi. Nämä asiat kannattaa kirjata ylös, jotta sääntöjen suunnittelu ja luonti ovat myöhemmin helpompaa. Mallintaminen kannattaa aloittaa, kun on selvillä, mitä osia suunnitteluautomaatti sisältää. Automaatilla ohjattavat mitat kannattaa nimetä, jolloin ne ovat helposti löydettävissä jälkeensä. Tämä helpottaa sääntöjen kirjoittamista.

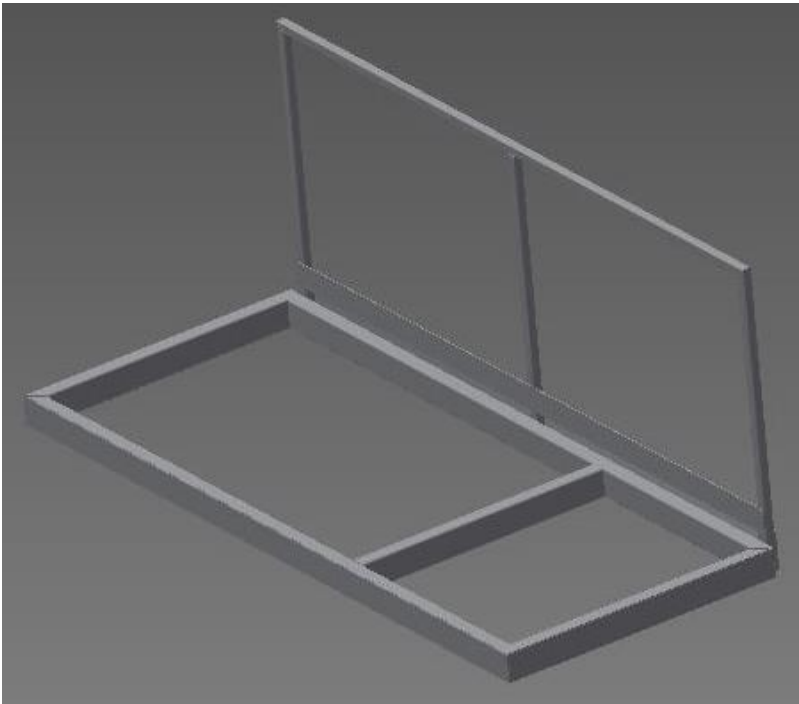
Tämä suunnitteluautomaatti sisältää kaikki tarvittavat osat ja deaktivoi ja aktivoi niitä tarpeen mukaan. Tämä on yksi syy miksi osa säännöistä on erittäin pitkiä. Toinen syy on itse iLogic, se ei tue kaikkia komentoja/piirteitä esim. component pattern -toiminnosta ei voida piilottaa elementtejä, vaan ne joudutaan piilottamaan osa kerrallaan.

6.1 Mallintaminen

Työssä tarvittu mallintaminen on yksinkertaista. Hyödyntämällä Inventor-toimintoja suunnitteluautomaatista saadaan helpommin ohjelmoitava. Näitä toimintoja on esim. Multi body -mallinnus. Mallintaminen aloitetaan luomalla ns. luuranko-, eli skeleton-malli. Tähän skeleton-malliin mallinnetaan kaikki automaatin osat. Kuvassa 10 on esitetty skeleton-malli. Kuvasta 10 on piilotettu ritilä osa, jotta kuva on havainnollisempi. Skeleton-mallilla tarkoitetaan osakuvaa, johon mallinnetaan kaikki halutut osat. Nämä osat luodaan uusina solideina skeleton-malliin, jolloin ne voidaan tallentaa omiksi osikseen kovalevyille. Alla on lista suunnitteluautomaattiin mallinnettavista osista ja kokoonpanoista.

- Osat
 - Runkopalkki
 - Tukipalkki
 - Kaiteen pystyputki
 - Kaiteen käsijohde
 - Kaiteen vaakaputki
 - Potkusta
 - Kaiteen kiinnitys holkki

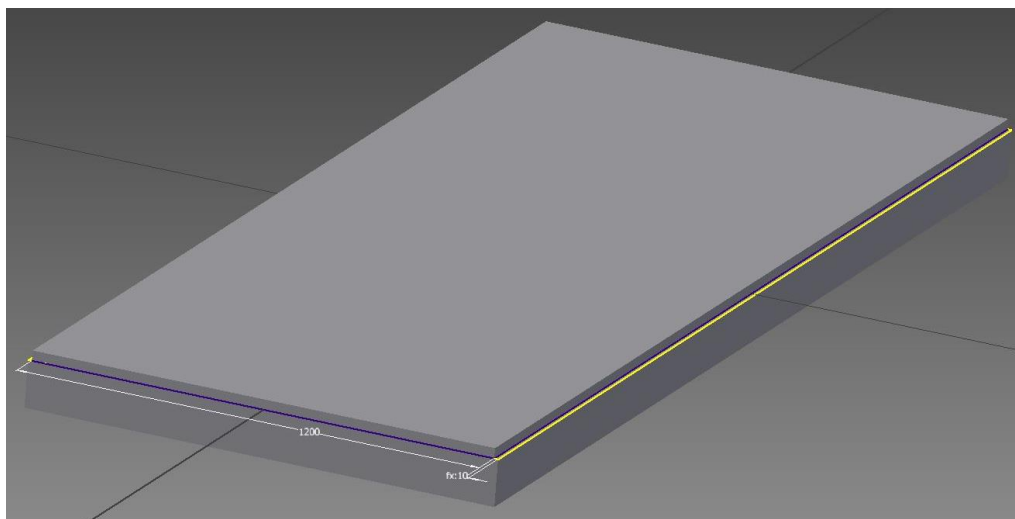
- Ritolätaso.
- Kokoonpanot
 - Rungonkokoonpano
 - Kaiteenkokoonpano
 - Loppukokoonpano.



KUVA 10. Skeleton-malli (Autodesk Inventor 2014)

6.1.1 Runko

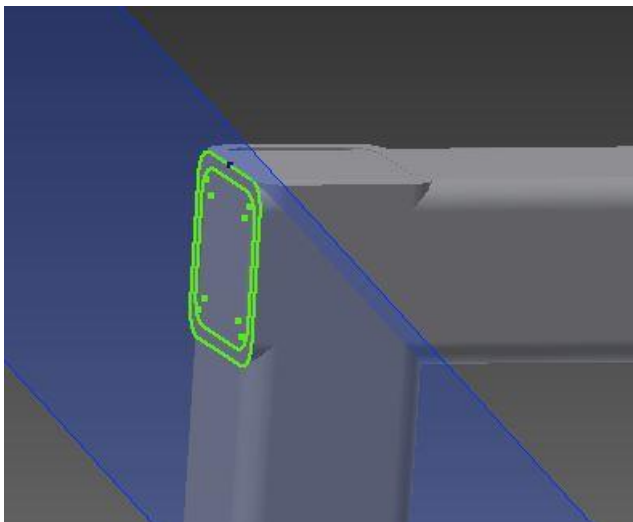
Rungon mallintaminen aloitetaan tekemällä suorakaide. Sen jälkeen suorakaide pursotetaan haluttuun korkeuteen. Tämän jälkeen rungon päälle mallinnetaan ritilä. Ritilän leveydellä tullaan ohjaamaan tason ja rungon leveyttä, joten rungon ja ritilän mitat kannattaa nimetä huolellisesti. Ritilä pursotetaan haluttuun mittaan ja nimetään. Kuvassa 11 näkyy rungon päälle piirretty sketch, josta ritilä on pursotettu omana solidina. Rungon solid:lla ei ole näkyvää tarkoitusta suunnitteluautomaatissa, vaan sitä käytetään runkopalkkien sijoittamiseen ja niiden sijainnin hallintaan. Rungolla myös hallitaan runkopalkkien korkeutta ja runkopalkkien pituutta.



KUVA 11. Runko ja ritilä (Autodesk Inventor 2014)

6.1.2 Kaiteet

Kaiteiden mallintaminen kannattaa aloittaa vasta, kun runko ja ritilä on mallinnettu. Kaiteiden mallinnus aloitetaan tekemällä sketch rungon pohjaan. Tähän sketchiin piirretään tukiholkki, pystyputki ja välipystyputki. Nämä voidaan myös tehdä omiin sketchiin. Pystyputken piirroksesta kannattaa luoda blokki, koska sitä tarvitaan uudelleen myöhemmin. Tämän jälkeen voidaan pursottaa sketchiin piirretyt osat omiksi solideiksi. Käsijohteen sketch tehdään pystyputken sivuun, jolloin käsijohde voidaan pursottaa pystyputken läpi. Kuvassa 12 näkyy, kuinka käsijohde on pursotettu pystyputken läpi. Käsijohde kannattaa pursottaa kaide elementin puoleenväliin asti ja peilata loppuosa, jolloin saadaan myös käsijohteen sculpt-toiminto peilattua. Sculpt-toiminnolla leikataan pystyputkesta ja käsijohteesta aputason avulla ylimenevät osat. Kuvassa 12 näkyy käsijohteen sketch ja aputaso, jolla käsijohde ja pystyputki leikataan. Pystyputki peilataan myös toiselle puolelle käyttäen käsijohteen puoliväliä, näin vältetään turhaa mallintamista. Kaiteen välijohde mallinnetaan pystyputken ja välipystyputken välille. Ainoastaan yksi välijohde on tarpeellista mallintaa.

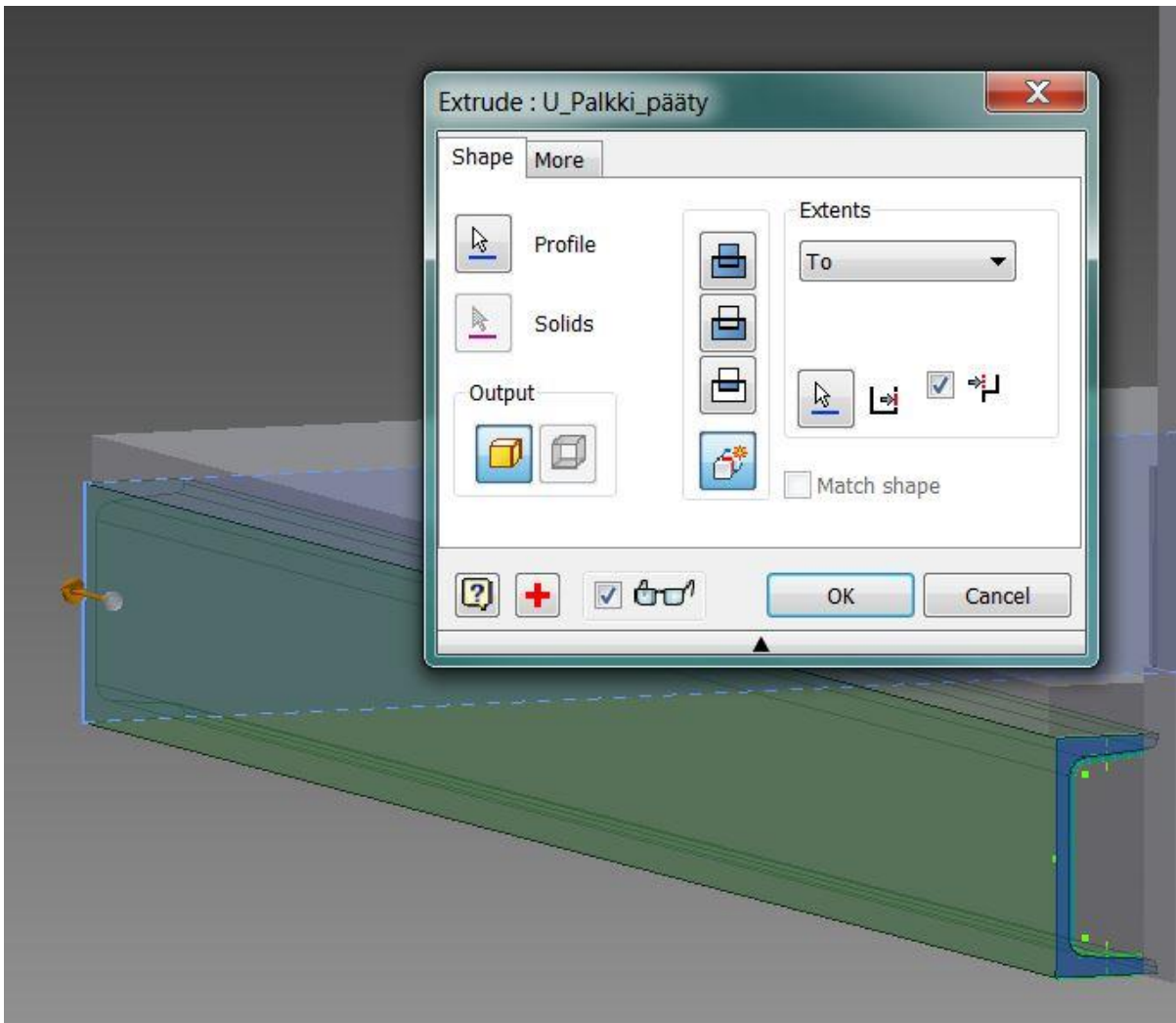


KUVA 12. Käsijohteen sketch ja pursotus (Autodesk Inventor 2014)

Putkikaiteet mallinnetaan lähes samalla tavalla kuin RHS-kaiteet. Poikkeuksina ovat esim. kulmat ja pursotukset. Potkulista mallinnetaan kaiteiden sisäpintaan eikä sen sijainnilla ei ole merkitystä, koska sijaintia tullaan muuttamaan myöhemmässä vaiheessa.

6.1.3 Runkopalkit

Runkopalkkien mallintaminen alkaa sketchin tekemisellä rungon kylkeen, tähän sketchiin piirretään runkopalkki. Runkopalkeille kannattaa tehdä omat sketchit, jotta niitä on helppo muuttaa jälkikäteen ja luoda niistä omat blokit. Kuvassa 13 esitetään, miten runkopalkit pursotetaan käyttämällä runkoa avuksi. U-palkin sketchiin luodaan aputaso, jolla U-palkki voidaan peilata toisin päin, jos U-palkin uuma halutaan ulospäin. Tämä aputaso tulee U-palkin puoleen väliin leveyssuunnassa. Runkopalkkien kulmat, jotka menevät päällekkäin leikataan sculpt-toiminnolla käyttäen aputasoja. Kuvassa 14 näkyy rungon leikkausta varten tehdyt kolme aputasoa. Kuvassa 14 näkyy myös, että runkopalkeista mallinnetaan vain toinen sivu ja toinen pääty. Näin säästetään mallintamiseen käytettyä aikaa ja helpotetaan sääntöjen kirjoittamista, koska muuttamalla alkuperäistä osaa muuttuu myös peilattu osa.



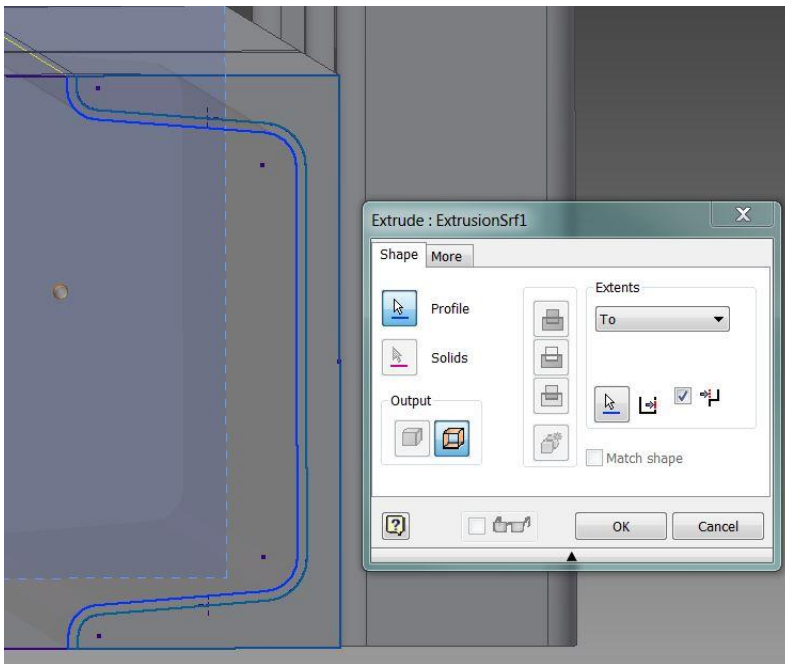
KUVA 13. Runkopalkin pursotus (Autodesk Inventor 2014)



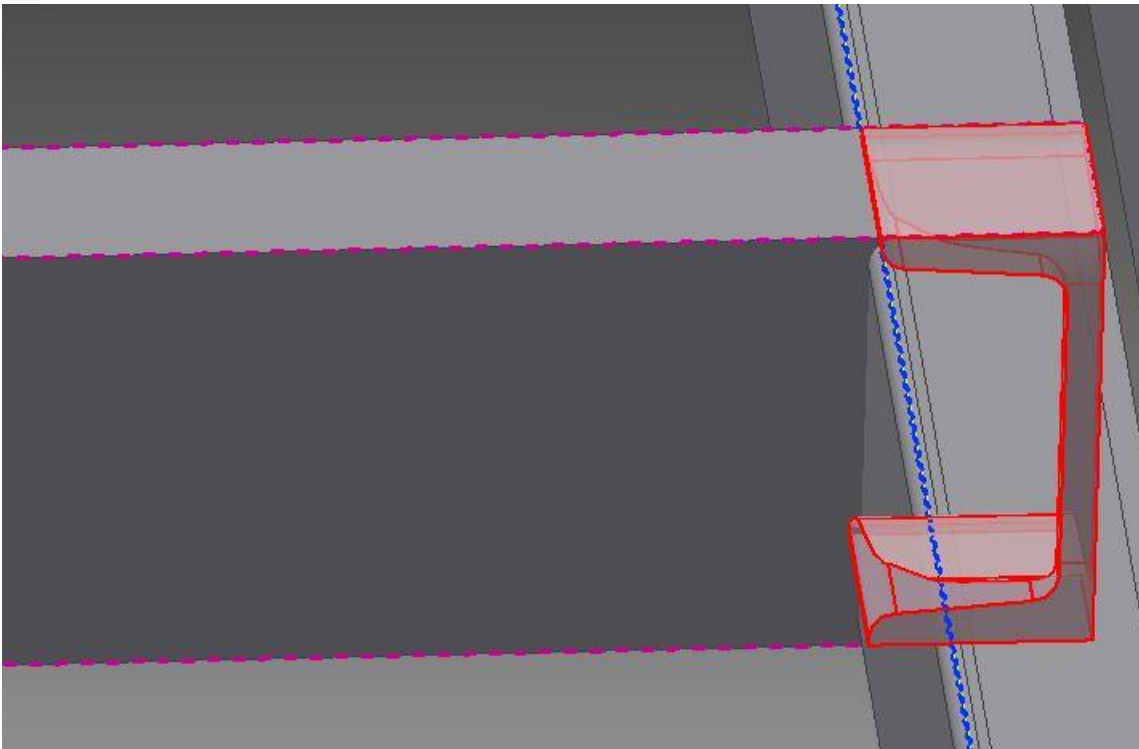
KUVA 14. Rungon aputasot (Autodesk Inventor 2014)

6.1.4 Tukipalkit

Tukipalkkien mallintaminen aloitetaan luomalla sketch tasolle, joka kulkee rungon läpi pitkittäis- ja pystysuunnassa origon kautta. Tälle Sketchille voidaan tuoda valmis blokki tai piirtää runkopalkki uudelleen. RHS-tukipalkki pursotetaan rungon sisäpintaan asti ja peilataan toiselle puolelle. RHS-tukipalkki voitaisiin myös piirtää runkopalkin pintaan ja pursottaa siitä vastakkaisen puolen runkopalkkiin asti. U-tukipalkki pursotetaan runkopalkin ulkopintaan asti, siitä leikataan ylimääräinen pituus. Toinen vaihtoehto on pursottaa U-palkki runkopalkin sisäpintaan ja sculpt toiminnolla pursottaa loppuosa. Pituus riippuu siitä, onko uuma sisään vai ulospäin. U-tukipalkin ja runkopalkin välille voidaan halutessa jättää rako. Kuvassa 15 näkyy, miten rako piirretään sketchiin tai blokkiin, tämä ei muuta blokin tai sketchin toimintaa. Kuvassa 16 runkopalkin uuma on sisäänpäin, jolloin U-tukipalkkia leikataan kuvassa 15 luodun surface:n avulla. Leikkaamisen jälkeen tukipalkki on peilaamista vaille valmis. Sculpt-toiminnolla suoritettun leikkauksen jälkeen tukipalkki voidaan peilata toiselle puolelle ja tukipalkki on valmis.



KUVA 15. Surfacen luonti (Autodesk Inventor 2014)



KUVA 16. U-tukipalkin sculpt (Autodesk Inventor 2014)

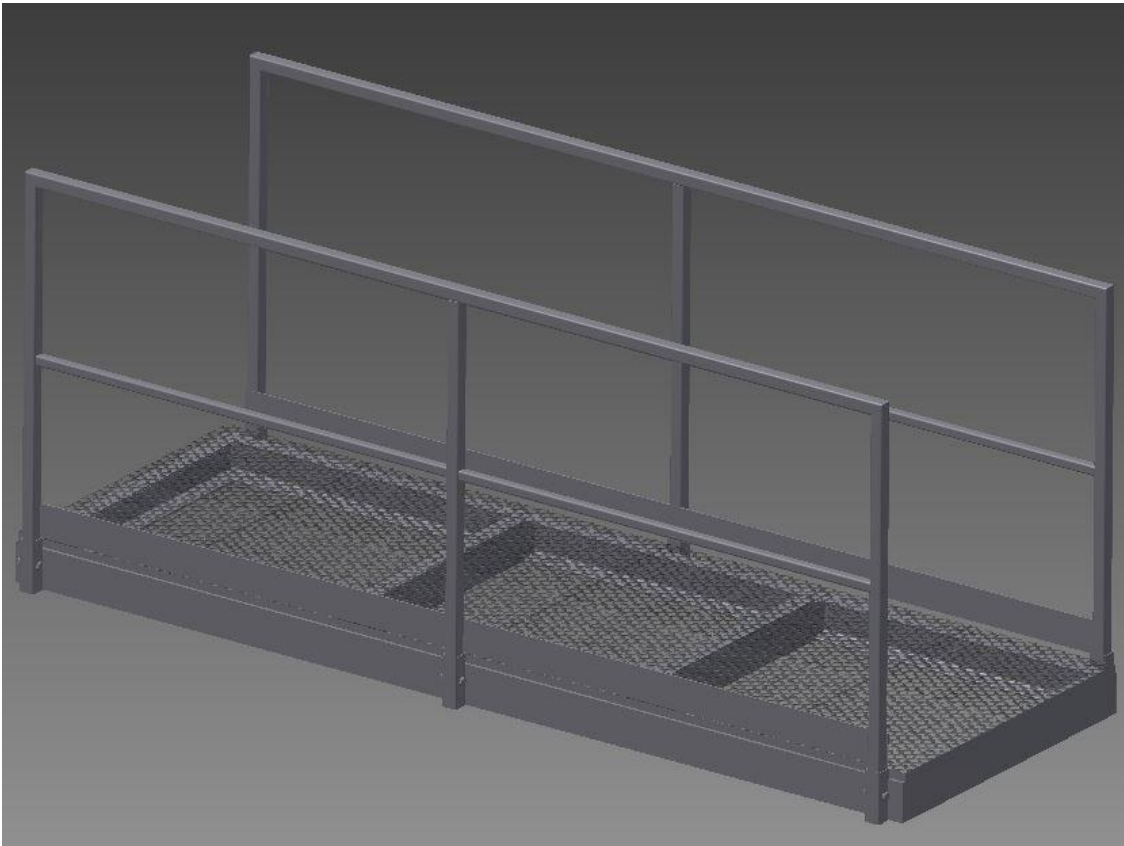
6.2 Kokoonpanot

Kokoonpano on nimensä mukaan kokonaisuus, johon tuodaan osia. Suunnitteluautomaatti sisältää kahdenlaisia kokoonpanoja, alikokoonpanoja ja pääkokoonpanon. Alikokoonpanoja tehdään kaksi kappaletta. Toinen tehdään rungosta ja toinen kaiteista. Näissä alikokoonpanoissa monistetaan tarvittut osat component pattern -toiminnolla. Näitä osia ovat esim. tukipalkit, tukiholkit, kaidevälijohde, potkulista ja kaide-elementti.

Pääkokoonpanoon kasataan viimeiset tarvittut osat ja alikokoonpanot yhteen. Myös skeleton-malli tuodaan pääkokoonpanoon, koska se mahdollistaa parametrien hakemisen helposti. Pääkokoonpano muodostuu neljästä eri asiasta. Nämä ovat kaiteiden alikokoonpano, rungon alikokoonpano, skeleton-malli ja ritilä. Pääkokoonpanoon lisätään vielä pultit ja mutterit Inventor content center –kirjastosta. Niitäkin monistetaan tarpeen mukaan component pattern toiminnolla. Skeleton-mallin näkyvyys pitää asettaa näkymättömäksi ja se pitää asettaa myös referenssi osaksi, jotta se ei näy mittakuvissa.

Kaikkien kokoonpanossa olevien osien level of detail pitää asettaa iLogic-tilaan, jos tällaista level of detailia ei ole olemassa pitää se luoda. Kuvassa 17 on valmis kokoonpano. Kokoonpanoihin tehtyjen

component pattern toimintojen ei tarvitse olla lopullisia, koska niitä tullaan myöhemmin ohjaamaan säännöillä.



KUVA 17. Kokoonpano (Autodesk Inventor 2014)

6.3 Säännöt

Sääntöjen kirjoittaminen on hidasta ja keskittymistä vaativaa työtä. Kirjoitettaessa sääntöjä kannattaa keskittyä yhteen asiaan kerralla, esim. runkoon tai kaiteisiin. Valmis sääntö kannattaa aina testata, jotta sen oikea toiminta muiden sääntöjen kanssa voidaan varmistaa. Autodesk:n forumilta ja muualta internetistä löytyy paljon muiden kirjoittamia sääntöjä. Niitä kannattaa hyödyntää mahdollisuuksien mukaan. Kuvassa 3 on iLogic:n käyttö liittymä, jolla säännöt luodaan.

6.3.1 Parametrien linkitys

Parametrien linkitys on yksi tärkeimpiä asioita, joita sääntöjen kirjoittamisessa tarvitaan; se mahdollistaa osien parametrien muuttamisen ja ohjaamisen. Tämän takia sitä kannattaa suunnitella etukäteen ja luoda lista tärkeistä parametreista. Kuvassa 18 on lista suunnitteluautomaatin tärkeistä parametreista ja niiden merkityksistä. Samalla tulee myös miettiä mitä tietoja mallia muuttavaan kaavakkeeseen halutaan, koska näille halutuille tiedoille pitää luoda omat parametrit. Parametreja voidaan kuitenkin luoda ja luultavasti tullaan luomaan lisää jälkikäteen.

Päämitat

- Pituus ohjaa tason pituutta, kaiteiden pituutta ja runkopalkkien pituutta
- Leveys ritilän leveys, osittain rungon leveys
- Kaiteet tuleeko kaide välilehteä, piilottaa tarpeettomat osat
- Ritilän tyyppi

Runko

- Palkkityyppi määrittää käytetyn profiilin ja niille tulevat leikkaukset yms.
- Profiilin suunta asettaa peilaus käskyn käyttöön esim. U-profiililla
- Profiilin päättyminen määrittää kiertäkö runko ympäri asti ja tuleeko profiilin päähän esim. laput kiinnitystä varten
- Ritilätason paksuus vaikuttaa kiinnitysholkin korkeuteen, kaiteen korkeuteen

Kaiteet

- Kaiteiden kiinnitys tuleeko kiinnitysholkkia, kaide kiinni runkoon
- Kaide profiilin valinta vaihtaa käytetyn profiilin, muuttaa kiinnitysholkin tyyppiä
- Kaide-elementtien määrä muuttaa kaiteen pituutta, kaidetolppien väliä (ilmoita yhden elementin pituuden)
- Kaiteiden sijainti vasen, oikea, molemmin puolin

KUVA 18. Tärkeitä parametreja ja niiden käyttökohteet

Parametrien linkitys tarkoittaa kahden parametrin yhdistämistä säännöllä, esim. parametri X = parametri Y. Parametrien linkittämisessä on tärkeä tietää, miten se toimii. Linkittäminen toimii siten, että yhtälön vasemmalla puolelle tulee ohjattava parametri ja oikealle puolelle ohjaava parametri/lauseke. Parametri voidaan kirjoittaa kahdella eri tavalla sääntöön, Parameter("osanumero", "parametrin nimi") tai Parametrin nimi. Parametrin nimi tapaa voidaan käyttää ainoastaan tilanteessa, jossa käytetty parametri on samassa tiedostossa kirjoitettavan säännön kanssa. Parameter("osanumero", "parametrin nimi") tapaa käytetään, kun parametri ei ole samassa tiedostossa säännön kanssa, vaan esim. alikokoonpanossa. Kuvassa 19 näkyy parametrien linkityssääntöjä ja parametrien molemmat kirjoitustavat. Osalle linkitettävistä parametreista täytyy luoda vastine parameters-valikkoon, varsinkin jos osan parametri halutaan linkittää kokoonpanoon.

```

□ '''Linkitetään parametrejä yhteen muiden kokoonpanojen ja skeletonnin kanssa
|
| '''skele
| Parameter("MR25-01-01-1-01:1", "Ritilän_leveys")=Leveys
| Parameter("MR25-01-01-1-01:1", "Pituus")=Pituus
| Parameter("MR25-01-01-1-01:1", "Ritilä_välituen_etäisyys_päästä")= Ritilä_tukiväli_etäisyys
| Parameter("MR25-01-01-1-01:1", "Ritilän_paksuus") = Ritilä_paksuus
|
□ '''runko
| Parameter("MR25-01-01-3-01:1", "Ritilä_tuki_etäisyys") = Ritilä_tukiväli_etäisyys
| Parameter("MR25-01-01-3-01:1", "Ritilä_tuki_määrä")=Ritilä_tuki_määrä
| Parameter("MR25-01-01-3-01:1", "Tukiholkki_patterointi")=Kaide_elementin_pituus + 100
| Parameter("MR25-01-01-3-01:1", "Tukiholkki_määrä")=Pystyputki_väli_määrä + 2
| Parameter("MR25-01-01-3-01:1", "Tukiholkki_etäisyys_leveys") = Parameter("MR25-01-01-1-01:1
| Pultti_patterointi_sivu = Parameter("MR25-01-01-1-01:1", "Rungon_leveys") + Parameter("MR25
| Pulttui_pattern = Kaide_elementin_pituus + 100
| Pultti_patterointi_määrä = Pystyputki_väli_määrä + 2

```

KUVA 19. Parametrien linkitys (Autodesk Inventor 2014)

6.3.2 Runko

Rungon sääntöjen kirjoittamien aloitetaan runkopalkkien näkyvyyttä muuttavalla säännöllä. Tällä säännöllä piilotetaan jompikumpi palkkityyppi näkyvistä. Sääntö muodostuu if-lauseesta ja try-catch-lauseesta. Kuvassa 20 on sääntö, jolla asetetaan U-palkkityypit näkyviin ja piilotetaan RHS-palkkityypit. Sääntö toimii siten että, kun if-lauseen määrittämä vaatimus täyttyy niin valitut osat ja toiminnot joko aktivoidaan tai deaktivoidaan. Try-catch-lauseella tarkistetaan, löytyvätkö lauseen sisällä olevat osat kokoonpanosta. Jos osa löytyy kokoonpanosta, se syötetään if-lauseeseen. iLogic laukaisee virheilmoituksen, jos if-lauseeseen on kirjoitettu osa, mutta se puuttuu kokoonpanosta. Tällainen virheilmoitus voidaan estää käyttämällä try-catch-lausetta. Try-catch-lausetta kannattaa käytetään tilanteissa, jossa osa on luotu component pattern -toiminnolla.

U-Palkkeja ohjataan myös säännöllä, joka muuttaa U-Palkin uuman suuntaa. Säännön toiminta perustuu piirrepuussa olevan piirteen deaktivoimiseen. Mallinnusvaiheessa U-Palkki on peilattu aputason avulla toisinpäin, kun tämä peilaus deaktivoidaan, kääntyy U-Palkki toisinpäin. Säännöksi muodostetaan yksinkertainen if-lause, kuten kuvassa 21 on esitetty. Tällä säännöllä muutetaan myös tukipalkin sovitusta.


```

If Runkopalkin_tyyppi = "U-Palkki 120x55" Then
Component.IsActive("MR25-01-01-1-10:1") = 1
Component.IsActive("MR25-01-01-1-11:1") = 1
Component.IsActive("MR25-01-01-1-16:1") = 1
Component.IsActive("MR25-01-01-1-17:1") = 1
Component.IsActive("MR25-01-01-1-14:1") = 0
Component.IsActive("MR25-01-01-1-13:1") = 0
Component.IsActive("MR25-01-01-1-12:1") = 0
Component.IsActive("MR25-01-01-1-15:1") = 0
Feature.IsActive("MR25-01-01-1-01:1", "Extrusion13") = 0
Feature.IsActive("MR25-01-01-1-01:1", "Extrusion14") = 1
Try
Component.IsActive("MR25-01-01-1-19:1") = 1
Component.IsActive("MR25-01-01-1-19:2") = 1
Component.IsActive("MR25-01-01-1-19:3") = 1
Component.IsActive("MR25-01-01-1-19:4") = 1
Component.IsActive("MR25-01-01-1-19:5") = 1
Component.IsActive("MR25-01-01-1-19:6") = 1
Component.IsActive("MR25-01-01-1-19:7") = 1
Component.IsActive("MR25-01-01-1-19:8") = 1
Component.IsActive("MR25-01-01-1-19:9") = 1
Component.IsActive("MR25-01-01-1-19:10") = 1
Component.IsActive("MR25-01-01-1-19:11") = 1
Component.IsActive("MR25-01-01-1-19:12") = 1
Component.IsActive("MR25-01-01-1-19:13") = 1
Component.IsActive("MR25-01-01-1-19:14") = 1
Component.IsActive("MR25-01-01-1-19:15") = 1
Component.IsActive("MR25-01-01-1-19:16") = 1
Component.IsActive("MR25-01-01-1-19:17") = 1
Component.IsActive("MR25-01-01-1-19:18") = 1
Component.IsActive("MR25-01-01-1-19:19") = 1
Component.IsActive("MR25-01-01-1-19:20") = 1
Component.IsActive("MR25-01-01-1-18:1") = 0
Component.IsActive("MR25-01-01-1-18:2") = 0
Component.IsActive("MR25-01-01-1-18:3") = 0
Component.IsActive("MR25-01-01-1-18:4") = 0
Component.IsActive("MR25-01-01-1-18:5") = 0
Component.IsActive("MR25-01-01-1-18:6") = 0
Component.IsActive("MR25-01-01-1-18:7") = 0
Component.IsActive("MR25-01-01-1-18:8") = 0
Component.IsActive("MR25-01-01-1-18:9") = 0
Component.IsActive("MR25-01-01-1-18:10") = 0
Component.IsActive("MR25-01-01-1-18:11") = 0
Component.IsActive("MR25-01-01-1-18:12") = 0
Component.IsActive("MR25-01-01-1-18:13") = 0
Component.IsActive("MR25-01-01-1-18:14") = 0
Component.IsActive("MR25-01-01-1-18:15") = 0
Component.IsActive("MR25-01-01-1-18:16") = 0
Component.IsActive("MR25-01-01-1-18:17") = 0
Component.IsActive("MR25-01-01-1-18:18") = 0
Component.IsActive("MR25-01-01-1-18:19") = 0
Component.IsActive("MR25-01-01-1-18:20") = 0
Catch
End Try
End If

```

KUVA 20. Palkkityyppi sääntö (Autodesk Inventor 2014)

```

'''2. Muutetaan U-palkin suuntaa
If Profiilin_suunta = "Sisään" Then
Feature.IsActive("MR25-01-01-1-01:1", "U_pääty_kääntö") = 0
Feature.IsActive("MR25-01-01-1-01:1", "U_sivu_kääntö") = 0
Feature.IsActive("MR25-01-01-1-01:1", "Sculpt9") = 1
Feature.IsActive("MR25-01-01-1-01:1", "Sculpt10") = 0
Else
Feature.IsActive("MR25-01-01-1-01:1", "U_pääty_kääntö") = 1
Feature.IsActive("MR25-01-01-1-01:1", "U_sivu_kääntö") = 1
Feature.IsActive("MR25-01-01-1-01:1", "Sculpt9") = 0
Feature.IsActive("MR25-01-01-1-01:1", "Sculpt10") = 1
End If

```

KUVA 21. U-Palkin suuntaa muuttava sääntö (Autodesk Inventor 2014)

6.3.3 Tukipalkit

Tukipalkkeja ohjaavia sääntöjä on kolme. Yksi ohjaa tukipalkkien määrää, toinen niiden etäisyyttä ja kolmas piilottaa tukipalkin. Tukipalkkien määrää muuttava sääntö on kaava, myös tukipalkkien väliä muuttava sääntö on kaava. Nämä kaavat voidaan kirjoittaa parametreihin tai sääntöihin. Tässä työssä se on kirjoitettu parametreihin, jolloin se päivittyy automaattisesti, kun parametrit muuttuvat.

$$\text{ritilätuki määrä} = \text{floor} \left(\frac{(\text{tason pituus} - \text{runkopalkin leveys} * 2)}{(\text{ritilän tukiväli} + \text{runkopalkin leveys})} \right)$$

Jossa Floor = alaspäin pyöristys

Ritilän tukiväli = Kahden tuen välinen suurin sallittu etäisyys

Ensimmäisellä säännöllä lasketaan ritilätukien määrä ja muutetaan sitä.

$$\text{ritilän tukiväli} = \frac{(\text{tason pituus} - \text{runkopalkin leveys})}{(\text{ritilätuki määrä} + 1)}$$

Toisella säännöllä lasketaan tukipalkkien väli ja muutetaan sitä.

Kolmannella säännöllä piilotetaan tukipalkit, jos runko on niin. Kuvassa 22 on tukipalkin piilottava sääntö. Säännöllä deaktivoidaan piirre, jolla tukipalkki tehdään.

```
If Ritilä_tuki_määrä = 0 Then
Feature.IsActive("MR25-01-01-1-01:1", "Extrusion13") = 0
Feature.IsActive("MR25-01-01-1-01:1", "Extrusion14") = 0
End If
```

KUVA 22. Tukipalkin deaktivoiva sääntö (Autodesk Inventor 2014)

6.3.4 Kaiteet

Kaiteita ohjataan useilla eri säännöillä. Näillä säännöillä hallitaan esim. kaiteiden määrää, pituutta, profiilia ja puolta. Kaiteiden sääntöjen kirjoittaminen aloitetaan säännöllä, jolla asetetaan kaiteiden osat aktiivisiksi. Tämän jälkeen muut säännöt voidaan kirjoittaa muotoon, joilla tarpeettomat deaktivoidaan pois, joissakin tapauksissa osia aktivoidaan tarpeen mukaan. Tällä tavalla vältetään helposti tehtäviltä ongelmilta esim. päällekkäisiltä säännöiltä, joista toinen deaktivoiva ja toinen aktivoi osan takaisin.

Kaiteiden tärkeimpiä sääntöjä on sääntö/kaava, jolla lasketaan montako välijohdetta ja välipystyputkea kaiteisiin tulee. Tämä määrää riippuu siitä, montako kaide elementtiä tasossa on ja kuinka pitkä taso on. Standardin mukaan pystyputkien suurin sallittu etäisyys on 1500mm. Kuvassa 23 on sääntö, jolla lasketaan pystyputkien määrä molemmilla profiileilla. Saatu tulos muutetaan yksiköttömäksi ja pyöristetään alaspäin, jotta component pattern toiminto ymmärtää sen. Välijohteiden määrä saadaan lisäämällä väliputkien määrään yksi. Pystyputkien lopulliset välit voidaan laskea, kun pystyputkien määrä on tiedossa

```
If Kaide_profiili = "RHS" Then
Välitolppien_määrä_lasku=Floor((Kaide_elementin_pituus- 2 * 40 mm)/(1500 + 40 mm))
Else If Kaide_profiili = "Putki" Then
Välitolppien_määrä_lasku=Floor((Kaide_elementin_pituus- 2 * 42.4 mm)/(1500 + 42.4 mm))
End If
```

KUVA 23. Pystyputkien määrän sääntö

$$\text{Kaide elementin pituus} = \frac{\text{Tason pituus} - 100\text{mm} * \text{kaide elementtien määrä}}{\text{kaide elementtien määrä}}$$

Jossa Kaide-elementtien määrä = Määritetään kaavakkeen parametrissa

$$\text{Pystytolppien etäisyys} = \frac{(\text{Kaide elementin pituus} - 40 * 2 - 40 * \text{Välitolppien määrä})}{(\text{Välitolppien määrä} + 1)}$$

Tästä kaavasta saadaan pystytolppien välinen etäisyys. Kyseinen kaava on RHS-profiilille.

$$\text{Pystytolppien etäisyys} = \frac{(\text{Kaide elementin pituus} - 42,4 * 2 - 42,4 * \text{Välitolppien määrä})}{(\text{Välitolppien määrä} + 1)}$$

Putki-profiilin kaava on lähes samanlainen, kuin RHS-profiilille. Ennen kuin saatua tulosta käytetään profiilien component pattern -toiminnossa, tulee siihen lisätä profiilin leveys, ja tämä tulee tehdä myös RHS-profiililla. Syy tähän on, että laskettu pystyputkien väli ei ole keskilinjojen väli.

Kaiteen etäisyyttä tason sivusta ohjataan if-lauseella. Säännöllä muutetaan sekä RHS- että putkiprofiilien etäisyyttä. Profiilien etäisyydet reunasta vaihtelevat niiden kokoeron takia, ja tämä pitää ottaa huomioon sääntöä kirjoittaessa. Kuvassa 24 neljä eri parametria, joita säännössä muutetaan, näistä kaksi on RHS-profiilille ja kaksi putkiprofiilille.

```

If Kaide_irroitettava="Kyllä" Then
Parameter("MR25-01-01-3-02:1", "Kaiteen_patterointi_leveys") = Parameter("MR25-01-01-1-01:1", "Rungon_1
Parameter("MR25-01-01-1-01:1", "Kaide_irroitettava")=Parameter("MR25-01-01-1-01:1", "Rungon_leveys")+5
Parameter("MR25-01-01-1-01:1", "Kaide_irroitettava_putki") = Parameter("MR25-01-01-1-01:1", "Kaide_putki
Parameter("MR25-01-01-3-02:1", "Kaiteen_putki_patterointi_leveys") = Parameter("MR25-01-01-1-01:1", "Ru
Else If Kaide_irroitettava="Ei" Then
Parameter("MR25-01-01-3-02:1", "Kaiteen_patterointi_leveys") = Parameter("MR25-01-01-1-01:1", "Rungon_1
Parameter("MR25-01-01-1-01:1", "Kaide_irroitettava")=Parameter("MR25-01-01-1-01:1", "Rungon_leveys")+0
Parameter("MR25-01-01-1-01:1", "Kaide_irroitettava_putki") = Parameter("MR25-01-01-1-01:1", "Kaide_putki
Parameter("MR25-01-01-3-02:1", "Kaiteen_putki_patterointi_leveys") = Parameter("MR25-01-01-1-01:1", "Ru
End If

```

KUVA 24. Kaiteen etäisyyttä reunasta muuttava sääntö (Autodesk Inventor 2014)

Seuraava kaiteita muuttava sääntö on kaiteen puolta muuttava sääntö. Tällä säännöllä hallitaan, minkä puolen kaide on näkyvissä. Tämä sääntö on erittäin pitkä, mutta yksinkertainen. Se deaktivoi kaiteen osat joita ei haluta nähdä. Säännöstä muodostui tässä työssä noin 100 riviä pitkä sääntö, ja se jouduttiin tekemään molemmille profiili tyypeille ja puolille. Kuvassa 25 on kaiteen puolta muuttavasta säännöstä pieni osa, ja siinä näkyvät vasemman puolen ensimmäistä kaide-elementti deaktivoidut toiminnot.

```

If Kaiteet = "Oikea" And Kaide_profiili = "RHS" Then
'Component Pattern 4:1 elementti 1 Component Pattern
Component.IsActive("MR25-01-01-1-08:2") = 0
Component.IsActive("MR25-01-01-1-07:2") = 0
Component.IsActive("MR25-01-01-1-03:2") = 0
Component.IsActive("MR25-01-01-1-05:2") = 0
Component.IsActive("MR25-01-01-1-09:9") = 0
Component.IsActive("MR25-01-01-1-09:10") = 0
Component.IsActive("MR25-01-01-1-09:11") = 0
Component.IsActive("MR25-01-01-1-09:12") = 0
Component.IsActive("MR25-01-01-1-09:13") = 0
Component.IsActive("MR25-01-01-1-09:14") = 0
Component.IsActive("MR25-01-01-1-09:15") = 0
Component.IsActive("MR25-01-01-1-09:16") = 0
Component.IsActive("MR25-01-01-1-06:8") = 0
Component.IsActive("MR25-01-01-1-06:9") = 0
Component.IsActive("MR25-01-01-1-06:10") = 0
Component.IsActive("MR25-01-01-1-06:11") = 0
Component.IsActive("MR25-01-01-1-06:12") = 0
Component.IsActive("MR25-01-01-1-06:13") = 0
Component.IsActive("MR25-01-01-1-06:14") = 0

```

KUVA 25. Kaiteen puolen sääntö (Autodesk Inventor 2014)

Kaiteen puolta ohjaavan säännön jälkeen tehdään sääntö, joka muuttaa kaide elementtien määrää. Kaide elementtien määrä valitaan kaavakkeessa olevasta parametreista, ja tämän parametrin mukaan sääntö poistaa käytöstä tarpeettomat kaide elementit. Säännöstä tehdään kuusi eri versiota, kuvassa 26 ja 27 näkyy kaksi eri versiota. Säännössä käytetään <= merkintää, sillä vähennetään säännön pituutta huomattavasti. Kyseisellä merkillä voidaan ajaa monta if-lausetta päällekkäin, kunhan vaatimukset täyttyvät. erimerkiksi. Kaide_elementtien_määrä parametri on 1, tällöin sekä kuvassa 26 ja 27 olevat if-lauseet toteutuvat ja molemmat kaide-elementit deaktivoidaan.

```

If Kaide_elementtien_määrä <= 3 ul And Kaide_profiili = "RHS" Then
Component.IsActive("MR25-01-01-1-08:7") = 0
Component.IsActive("MR25-01-01-1-08:8") = 0
Component.IsActive("MR25-01-01-1-07:7") = 0
Component.IsActive("MR25-01-01-1-03:7") = 0
Component.IsActive("MR25-01-01-1-05:7") = 0

```

KUVA 26. Kaide elementin deaktivointi (Autodesk Inventor 2014)

```

If Kaide_elementtien_määrä <= 2 ul And Kaide_profiili = "RHS" Then
Component.IsActive("MR25-01-01-1-08:5") = 0
Component.IsActive("MR25-01-01-1-08:6") = 0
Component.IsActive("MR25-01-01-1-07:5") = 0
Component.IsActive("MR25-01-01-1-03:5") = 0
Component.IsActive("MR25-01-01-1-05:5") = 0

```

KUVA 27. Kaide elementin deaktivointi (Autodesk Inventor 2014)

Välitolppien ja välilohteiden määrä on selvillä, joten voidaan luoda sääntö, joka deaktivoi ylimääräiset välitolpat ja välilohteet. Tämä sääntö toimii samalla tavalla kaide elementtejä deaktivoivan säännön kanssa. Tästä säännöstä tehdään 14 versiota, seitsemän RHS-profiilille ja seitsemän putkiprofiilille. Kuvassa 28 on neljän kaide elementin ylimääräiset osat, jos välitolppia on kuusi. Välitolppia varten pitää

tehdä myös toinen sääntö. Tällä säännöllä deaktivoidaan välitolppien pursotus piirre, kuten kuvassa 29 näkyy. Tätä sääntöä tarvitaan, jos kaide on erittäin lyhyt, jolloin ei tarvita välitolppaa. Säännöstä tehdään molemmille profiileille omat versiot.

```
If Välitolppien_määrä <= 6 And Kaide_profiili = "RHS" Then
  'rhs
  Component.IsActive("MR25-01-01-1-09:56") = 0
  Component.IsActive("MR25-01-01-1-06:49") = 0
  Component.IsActive("MR25-01-01-1-09:64") = 0
  Component.IsActive("MR25-01-01-1-06:56") = 0
  Component.IsActive("MR25-01-01-1-09:40") = 0
  Component.IsActive("MR25-01-01-1-06:35") = 0
  Component.IsActive("MR25-01-01-1-09:48") = 0
  Component.IsActive("MR25-01-01-1-06:42") = 0
  Component.IsActive("MR25-01-01-1-09:24") = 0
  Component.IsActive("MR25-01-01-1-06:21") = 0
  Component.IsActive("MR25-01-01-1-09:32") = 0
  Component.IsActive("MR25-01-01-1-06:28") = 0
  Component.IsActive("MR25-01-01-1-09:16") = 0
  Component.IsActive("MR25-01-01-1-06:14") = 0
  Component.IsActive("MR25-01-01-1-09:8") = 0
  Component.IsActive("MR25-01-01-1-06:7") = 0
End If
```

KUVA 28. Välitolppien ja välijohteiden deaktivointi (Autodesk Inventor 2014)

```
If Välitolppien_määrä= 0 And Kaide_profiili = "RHS" Then
  Feature.IsActive("MR25-01-01-1-01:1", "Kaide_väli_pystyputki") = 0
Else
  Feature.IsActive("MR25-01-01-1-01:1", "Kaide_väli_pystyputki") = 1
End If
```

KUVA 29. Välitolpan pursotusta muuttava sääntö (Autodesk Inventor 2014)

Kaideen profiilia muuttavalla säännöllä deaktivoidaan toinen profiilityyppi. Tässä säännössä luetellaan kaikki parametria vastaamattoman profiilityypin osat ja deaktivoidaan ne. Kuvassa 30 on pieni osa säännöstä.


```

If Kaide_profiili = "Putki" Then
  'Component Pattern 4:1 elementti 1 Component Pattern 3:1 elementti 1 oikea kaide
  Component.IsActive("MR25-01-01-1-07:1") = 0
  Component.IsActive("MR25-01-01-1-03:1") = 0
  Component.IsActive("MR25-01-01-1-05:1") = 0
  Component.IsActive("MR25-01-01-1-09:1") = 0
  Component.IsActive("MR25-01-01-1-09:2") = 0
  Component.IsActive("MR25-01-01-1-09:3") = 0
  Component.IsActive("MR25-01-01-1-09:4") = 0
  Component.IsActive("MR25-01-01-1-09:5") = 0
  Component.IsActive("MR25-01-01-1-09:6") = 0
  Component.IsActive("MR25-01-01-1-09:7") = 0
  Component.IsActive("MR25-01-01-1-09:8") = 0
  Component.IsActive("MR25-01-01-1-06:1") = 0
  Component.IsActive("MR25-01-01-1-06:2") = 0
  Component.IsActive("MR25-01-01-1-06:3") = 0
  Component.IsActive("MR25-01-01-1-06:4") = 0
  Component.IsActive("MR25-01-01-1-06:5") = 0
  Component.IsActive("MR25-01-01-1-06:6") = 0
  Component.IsActive("MR25-01-01-1-06:7") = 0

```

KUVA 30. Kaiteen profiilin sääntö (Autodesk Inventor 2014)

6.3.5 Potkulista

Potkulistaa joudutaan ohjaamaan muutamalla säännöllä, joilla hallitaan sen sijaintia ja määrää. Kuvassa 31 on potkulistan sijaintia muuttavia sääntöjä. Ensimmäisellä if-lausekkeella kuvassa 31 muutetaan potkulistan ja tason keskilinjan välistä etäisyyttä. Tämä etäisyys vaihtelee sen mukaan, onko kaide irrotettava ja mikä profiilityyppi on käytössä. Potkulistan etäisyyttä hallitaan myös tason pituussuunnassa. Tällä tavalla potkulista saadaan pidettyä oikeassa kohdassa, vaikka profiilityyppejä muutettaisiin. Tällä myös estetään potkulistan sijainnin vääristyminen, kun suunnitteluautomaattia käytetään. Kuvan 30 alalaidassa näkyy pituus suunnan sijaintia hallitseva sääntö. Säännössä näkyy myös profiilien aiheuttama ero. Potkulistan sijainti Z akselilla on helppo asettaa ritilän pinnasta 10 mm ylöspäin.

```

If Kaide_profiili = "RHS" And Kaide_irroitettava = "Kyllä" Then
Parameter("MR25-01-01-3-02:1", "Potkulista_etäisyys_keskeltä") = Leveys /2 +10
Parameter("MR25-01-01-3-02:1", "potkulista_etäisyys_keskeltä2") = -(Leveys /2 +10)
Else If Kaide_profiili = "RHS" And Kaide_irroitettava = "Ei" Then
Parameter("MR25-01-01-3-02:1", "Potkulista_etäisyys_keskeltä") = Leveys /2 +5
Parameter("MR25-01-01-3-02:1", "potkulista_etäisyys_keskeltä2") = -(Leveys /2 +5)
End If

If Kaide_profiili = "Putki" And Kaide_irroitettava = "Kyllä" Then
Parameter("MR25-01-01-3-02:1", "Potkulista_etäisyys_keskeltä") = Leveys /2 +8.8
Parameter("MR25-01-01-3-02:1", "potkulista_etäisyys_keskeltä2") = -(Leveys /2 +8.8)
Else If Kaide_profiili = "Putki" And Kaide_irroitettava = "Ei" Then
Parameter("MR25-01-01-3-02:1", "Potkulista_etäisyys_keskeltä") = Leveys /2 + 5
Parameter("MR25-01-01-3-02:1", "potkulista_etäisyys_keskeltä2") = -(Leveys /2 + 5)
End If

'määritetään potkulistan sijainti pituus suunnassa
If Kaide_profiili = "Putki" Then
Parameter("MR25-01-01-3-02:1", "potkulistan_etäisyys_keskeltä_pituus") = -(Pituus/2-48.8)
End If
If Kaide_profiili = "RHS"
Parameter("MR25-01-01-3-02:1", "potkulistan_etäisyys_keskeltä_pituus") = -(Pituus/2-50)
End If

```

KUVA 31. Potkulistan sääntöjä (Autodesk Inventor 2014)

6.3.6 Muita sääntöjä

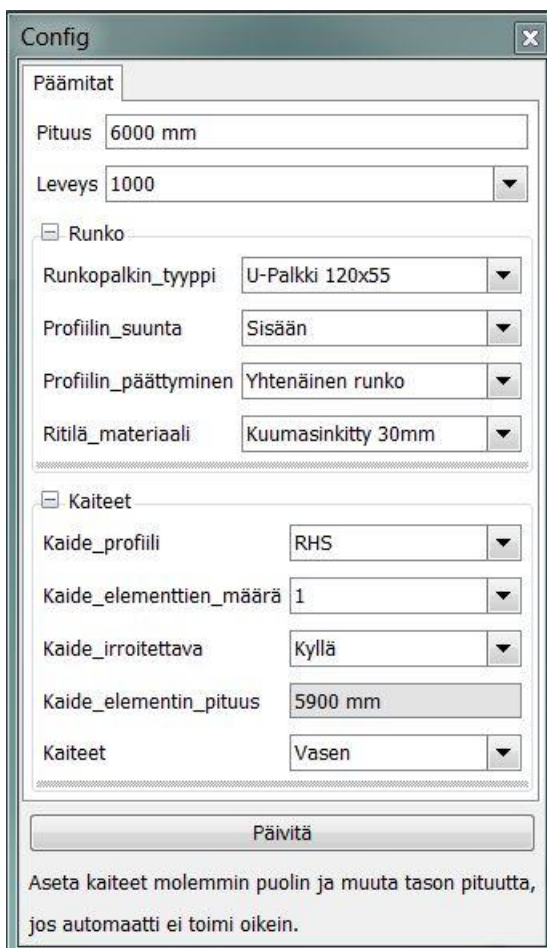
Suunnitteluautomaattiin tarvitaan myös muitakin kuin parametreja ohjaavia sääntöjä. Tällaisia ovat kaavakkeen laukaiseva sääntö ja kaavakkeessa käytettävä päivityssääntö.

Kaavakkeen laukaisevaa sääntöä käytetään helpottamaan ja yksinkertaistamaan suunnitteluautomaatin käyttöä. Kaavake laukeaa, kun suunnitteluautomaatti avataan, tämä onnistuu event triggerin ansiosta. Sillä pystytään määrittämään, että jokin sääntö ajetaan kun esim. uusi dokumentti luodaan. Kuvassa 32 näkyy. Sääntö itse on erittäin yksinkertainen lause **iLogicForm.Show("Config")**.

Päivityssäännöllä ohjataan järjestystä, missä muita sääntöjä ajetaan suunnitteluautomaatissa. Se sisältää myös lauseet, joilla päivitetään level of detail ja muutetaan skeleton-malli näkymättömäksi. Päivityssääntöä käytetään kaavakkeen kautta. Kuvassa 33 näkyy kaavakkeessa oleva painike päivityssäännölle.



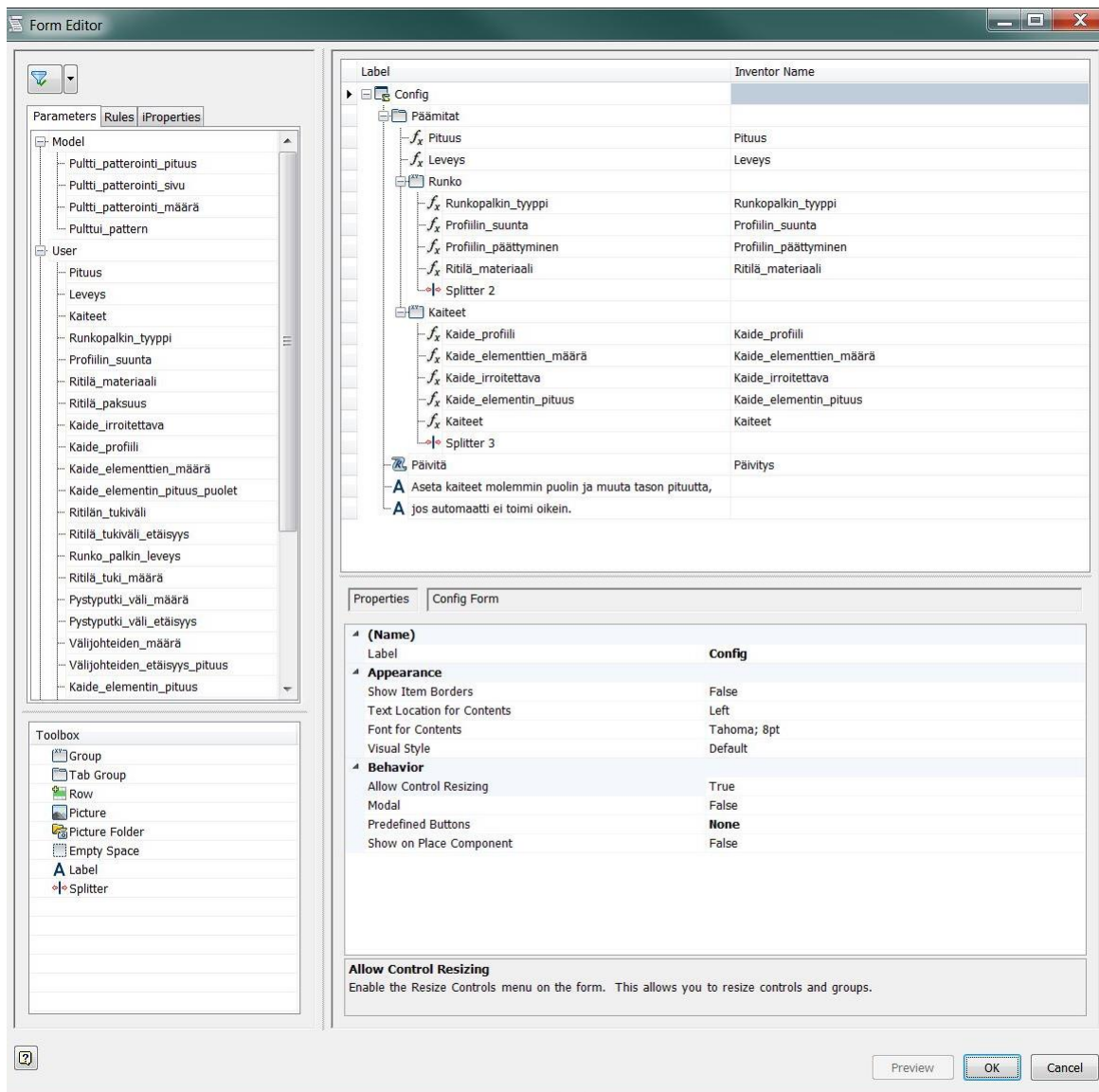
KUVA 32. Event trigger (Autodesk Inventor 2014)



KUVA 33. Kaavake (Autodesk Inventor 2014)

6.4 Kaavake

Kaavake on suunnitteluautomaatin eniten näkyvillä olevia osa, joten se kannattaa tehdä mahdollisimman yksinkertaiseksi ja selkeäksi käyttöä. Kaavake luodaan Manage-sivulta iLogic kohdasta; tarkemmin add form -kohdasta. Kaavake voidaan luoda joko paikalliseksi tai globaaliksi. Tässä tapauksessa se luodaan paikalliseksi. Kaavakkeen luonnin jälkeen avautuu kuvan 34 näköinen ikkuna, joka on kaavakkeen editori. Tällä kaavakkeella luodaan haluttu kaavake. Editorissa on 4 eri aluetta, vasemmassa yläkulmassa ovat parametrit, säännöt ja ipropertyt, oikealla yläkulmassa rakennepuu, vasemmassa alakulmassa työkalut ja oikealla alakulmassa asetukset. Editorissa näkyy esikatselu kaavakkeesta samalla kun kaavaketta muokataan, ja se näyttää samalta kuin kuvassa 33 oleva kaavake.



KUVA 34. Kaavake editori (Autodesk Inventor 2014)

7 POHDINTA

Opinnäytetyön tavoitteena oli suunnitella ja valmistaa hoitotasojen suunnitteluautomaatti. Tämä tavoite onnistuttiin täyttämään, lukuunottamatta mittakuvia. Mittakuvien automatisointi ei onnistunut, koska Inventor ei tarjonnut mahdollisia työkaluja sen tekemiseen. Itse kuvannot olisi pystytty luomaan automaattisesti, mutta mitat ja osapallot olisi pitänyt luoda itse kuviin. Tämä ongelma olisi ehkä pystytty ratkaisemaan jollakin tavalla, mutta rajallisen ajan takia en löytänyt toimivaa ratkaisua. Tästä syystä mittakuvia ei sisällytetty lopulliseen työhön. Työn aloituspalaverissa keskusteltiin hoitotason lujuustarkasteluista, ja ne myös kirjattiin työn yhdeksi tavoitteeksi, mutta suunnitteluautomaatin suuren työmäärän ja monien ongelmien takia osa lujuustarkasteluista unohtui tehdä.

Työn tekemisessä hyödynsin lähes kaikkia luovan työn tekniikoita, ja niistä oli suurta hyötyä opinnäytetyössä. Alku kankeuden ja tieto taidon puutteen vuoksi jouduin keräämään paljon tietoa iLogic:sta ja siitä, miten sitä voidaan käyttää opinnäytetyössäni. Sen jälkeen jouduin karsimaan paljon hakemaani tietoa, koska osa ei ollut hyödyllistä opinnäytetyössäni. Kun tietoa oli karsittu riittävästi, piti sitä myös pystyä hyödyntämään. Tässä auttoi useat esimerkit ja videot, joita löytyi yllättävänkin paljon muuhun teoriaan verrattuna. Tällainen tiedon haun ja jalostamisen kierre työn kannalta lähes pakollinen, tieto taidon puutteen vuoksi.

Työn tekovaiheessa esille tuli useita ongelmia. Nämä ongelmat johtuivat usein joko omasta tietämättömyydestä ja taitamattomuudesta tai iLogic:n puutteellisesta toiminnasta. Yksi suurimmista ongelmista oli component pattern -toiminnon ja iLogic:n yhteensopivuus. iLogic:lla pystytään ohjaamaan, kuinka monta osaa tai kokonaisuutta monistetaan component pattern toiminnolla. iLogic:lla ei voida ohjata component pattern toiminnon luomia kokonaisuuksia yhdellä kertaa, vaan näitä kokonaisuuksia pitää hallita sen sisältämien osien kautta.

Työ olisi voitu tehdä yksinkertaisemmin hyödyntämällä yksinkertaisia Visual Basic -koodeja tehokkaammin. Tällä tavalla olisi voitu hyödyntää tehokkaammin Inventorin käskyjä ja varsinkin component pattern toimintoa. Tämä olisi helpottanut työn tekemistä huomattavasti, koska component pattern toiminto on työssä eniten käytetty ja tärkein sääntöjä suunnitteluautomaatin toiminnan kannalta.

Visual Basic koodien tehokas hyödyntäminen vaatii kuitenkin tietämystä iLogic:sta ja Visual Basicista. Samalla pitää myös ymmärtää milloin yksinkertainen sääntö on helpompi ja nopeampi toteuttaa, kuin

monimutkainen koodi. Itselläni ei ollut ennestään kokemusta Visual Basicista muuta kuin Excelin kautta, ja tämäkin kokemus oli pääasiassa if-lauseen käyttöä. Tästä kokemuksesta oli kuitenkin hyötyä, koska pystyin if-lauseilla luomaan erittäin yksinkertaisia ja toimivia sääntöjä. Työn loppua kohden osasin jo käyttää useampaa Visual Basic -sääntöä tehokkaasti hyödyksi. Tämän ansiosta pystyin kirjoittamaan lyhempiä sääntöjä ja ratkaisemaan ongelmia, jotka tulivat työn alkuvaiheessa vastaan.

Jos nyt aloittaisin työn uudelleen, suunnittelisin automaatin hierarkian tarkemmin ja käyttäisin enemmän aikaa koodin ja automaatin jakamiseen pienempiin osiin. Tällöin automaattia olisi helpompi testata ja ongelmat helpompi paikantaa. Aikaisemman kokemuksen ansiosta osaisin ennakoida ongelmia, mitä joidenkin Inventorin käskyjen ja iLogic:n välillä on. Suurimpana hyötynä olisi silti oppimisen kautta saatu kokemus ja tieto, miten Inventorin käskyt toimivat. Normaalisti ei ole väliä millä tavalla component pattern toimintoa käytetään, kunhan saadaan tuotettua haluttu lopputulos. iLogic:n kanssa on tärkeää, miten päin rivit ja sarakkeet valitaan, kun käytetään component pattern toimintoa. Rivien ja sarakkeiden valinta component pattern toiminnossa vaikuttaa osien järjestykseen rakennepuussa ja samalla niiden nimeämiseen.

Työn alkuvaiheessa tehdyt päätökset ja tavoitteet vaikuttivat työn tekemiseen ja edistymiseen: sekä kannustavasti, että vähän rajoittavasti. Ne eivät kuitenkaan sulkeneet mitään ovia, vaan ennemminkin asettivat suunnan, mitä päin oli tarkoitus lähteä etenemään. Vaikka eteneminen työssä ei ollut kokoajan helppoa ja nopeaa, ei tämä siltikään tarkoittanut, ettei työssä voinut edetä eteenpäin. Välillä jokin ongelma vaati enemmän aikaa, jotta se pystyttiin ratkaisemaan. Joitakin ongelmia en saanut ratkaistua, vaan jouduin kiertämään ne esim. palaamalla taaksepäin työssä ja tekemällä asian eri tavalla. Tämä oli usein haasteellista ja aikaa vievää, mutta samalla myös todella palkitsevaa.

Opinnäytetyön kirjoittamisessa haasteellisinta oli tekstin tuottaminen ja kerrontatavan valitseminen, koska tämä työ ei ole ohje vaan ennemminkin dokumentti siitä, miten suunnitteluautomaatti voidaan toteuttaa. Työssä voitaisiin kertoa jokaisesta osa-alueesta enemmän, mutta tällöin työn kerrontatapa muuttuisi enemmän ohjeen muotoon. Tekstin tuottamisen ongelma on itselläni pääasiassa häiriötekijöistä johtuva, joten tässä ongelmassa auttoi, että sain tehdä opinnäytetyötä Merius Oy:llä.

Työn suurimpia ongelmia on omasta mielestäni teoreettisen tiedon puute, mutta hankintaan en pystynyt käyttämään rajattomasti aikaa. Aiheen vierestä olisi kyllä löytynyt tietoa, vaikka millä mitalla.

Kokonaisuutta ja työn laajuutta ajatellen onnistuin opinnäytetyössäni hyvin. Kehitettäviä paikkoja tietenkin aina löytyy.

LÄHTEET

Autodesk Inventor Release History. 2015. Saatavissa <http://www.cadprofessor.in/2008/11/autodesk-inventor-release-history/>. Viitattu 9.5.2016.

Konedirektiivi 2006/42/EY. 17.5.2006. Saatavissa <http://eur-lex.europa.eu/legal-content/fi/ALL/?uri=CELEX:32006L0042>. Viitattu 17.2.2016

Luovan työn opas. FilosofianAkademia. 2009. Saatavissa <http://docplayer.fi/314295-Luovan-tyon-opas-1-0-www-filosofianakatemia-fi-informaatio-filosofianakatemia-fi.html>. Viitattu 19.5.2016.

SFS-EN ISO 14122-1. KONETURVALLISUUS. 2010. Helsinki: Suomen standardisoimisliitto SFS.

SFS-EN ISO 14122-2. KONETURVALLISUUS. 2010. Helsinki: Suomen standardisoimisliitto SFS.

SFS-EN ISO 14122-3. KONETURVALLISUUS. 2010. Helsinki: Suomen standardisoimisliitto SFS.

Ympäristöministeriön asetus rakennusten paloturvallisuudesta. 6.4.2011. Saatavissa http://www.finlex.fi/data/normit/37126-E1_2011-fi.pdf. Viitattu 9.5.2016.